The
Small Computer
Magazine

kilobaud™

Understandable for beginners, Interesting for experts

January 1978 / Issue #13 / $2.00 / DM7,50 / Sfr8,10 / Ffr16,0

SWTPC NEWSLETTER

6800/2 Announced

Oct. 24, 1977 Southwest Technical Products Corp. today announced the immediate availability of the 6800/2 Computer System. The 6800/2 features the new A2 processor board which incorporates sockets for 8K of PROM/EPROM and which uses the new MC6875 monolythic clock driver. DIP switches are provided to allow any 8K block of upper memory (above 32K) to be switched to external RAM or EPROM. This makes it easy to place process control, custom monitors or ROM BASIC in the system.

The MC6875 clock oscillator is independent of the baud rate generator and its operating frequency may be adjusted by changing the value of a single resistor. This allows easy use of higher clock rates if the user wishes to convert the system to premium processor and memory components.

The 6800/2 will be supplied with the new MP-A2 processor board, 4K of memory on an MP-8M board and with a MP-S serial interface. The complete kit with power supply and case is priced at **$439.00** postpaid in Continental U.S.

Dear 6800 Users,

This newsletter—ad is being tried in response to many requests for a better form of communication between Sw. Tech. and 6800 System users. We have tried in the past to send information on new accessories and software directly to customers and to our dealers, but many of you are evidently not getting these as you should.

We plan to periodically make up an ad like this one and tell you about any new developements. If you want more information on any of these items please write and let us know. We will be happy to send it. Our new computer products catalog describing the new products presented here is now available.

SWTBUG®

If you are still using Mikbug® as a monitor in your 6800 System you should consider replacing it with our new SWTBUG® monitor. It is a direct replacement and hardware compatible **without** any system modifications. Features are:

* Automatic Disk Boot—just type **D**
* Automatic interface configuration for either MP-S or MP-C
* Vectored Software Interrupt
* Single Level Breakpoints
* Punch End of Tape Formatting
* Subroutine Entry Point Compatibility

Odds and Ends

Some of you out there may not have heard about our MP-T Interrupt Timer interface. This is a software programmable timer with selectable interrupts from 1 microsecond to 1 hour.

"Number Nibbler" is what we call the MP-N Calculator Interface. This handy little device makes it easy to do arithmetic functions when they are part of a machine language, or assembler program.

The listing for SWTBUG® is supplied with the monitor. We can also provide listings of our DOS as used in the MP-F Minifloppy for those who are interested in incorporating special features. Since this is not a volume item, we will copy these on an individual basis for those interested. Price will be $10.00.

Sneaky Change of the Month—Last summer we made a small change in the MP-L Parallel Interface. It is now officially designated MP-LA. Full buffering of **all** outputs and inputs was added as well as the ability to operate with 16 bits out.

16K Memory

Southwest Technical Products Corp. is offering for late Dec. or early Jan. shipment 16K Memory Boards—assembled and tested for only $400.00. These memory boards make it possible to fully expand the RAM memory of the 6800 System with only one memory board. The 16K of dynamic memory supplied with the board can be expanded to 32K by simply soldering an additional sixteen IC's in place. The use of dynamic memory in no way slows the processor speed when these memories are used. Memory refresh is accomplished during the second clock phase while the processor is idle.

MP-16 Memory—assembled and tested . . . . .ppd in Continental U.S. $400.00
16-EX Expansion Kit. ppd Cont. U.S. $250.00

EPROM Programmer

Now not only can you easily add EPROM's to your 6800 System, but you can also program them yourself right in the machine. The MP-R allows you to program and to verify 2716 EPROMs with no additional equipment necessary. (2716s using +5 Volt **only** power supplies—Intel, etc.) The MP-R plugs into an interface slot and the EPROM socket extends above the top of the case for easy access. All software necessary to operate the programmer is supplied with the kit and includes provision for verifying, testing and copying 2716 EPROMS. The programming voltage is generated by an on-board DC to DC converter. Only the normal supply voltage from the 6800 supply is needed.

MP-R Programmer Kit with software . . . . ppd in Continental U.S. $44.95

SOON—

Assembled and tested 6800 systems. Write or contact your nearby SwTPC dealer.

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. Rhapsody
San Antonio, Texas 78216

PUBLISHER'S REMARKS

Houston Scene

The Houston computerfest drew a good crowd – almost all hobbyists – because it was largely promoted for this group. Many of the exhibits (about 35) were local dealers – plus a few manufacturers – some parts dealers such as Jade, S.D. Sales, Tri-Tek, and only two magazines . . . *Kilobaud* and *Personal Computing*.

This was my first chance to take a good look at the Noval system (Photo 1), and it made me want to have someone who owns one give us an evaluation. It certainly is attractive and seems complete, but until we've had an opportunity to check it out as a system, complete with software, it's difficult to really tell. TEI made a good impression too . . . as they did at Computermania. Here's president Garry Walker (Photo 2). TEI certainly had one of the more professional-looking exhibits.

Chuck Watson of Tri-Tek (Photo 3) was kept busy selling parts, as was Jim Tanner of DRC (Photo 4). The lack of packed-in people made it possible for attendees to see the exhibits closely and ask questions (Photo 5).

Dealers had more of a chance to sift through the hobbyists for good customer prospects . . . here's Rick Spitler of Micro Vision (Photo 6). Despite being ripped off by a former employee of a good deal of his store stock, Bill Rogers (Photo 7) of Interactive Computers is busy opening a second store . . . with more in the works. Ray Atnip of the Bit Barn (Photo 8) was selling floppy disks and T-shirts. He said the computerfest did very well for him. Jack McKinstry (Photo 9) has just recently opened a Houston Computerland and was also happy about the exposure from the show. Bob Seydler (Photo 10) was there for the Houston Computer Mart.

John Barugh of Texas Micro Games (Photo 11) was there with microcomputerized backgammon and couldn't keep people away from his booth. Richard Liedler (Photo 12) of Salt Lake City was there with his parents and monopolized many of the games being demonstrated. This show, like most hobby computer shows, was heavy on games . . . very heavy.

One of the complaints about the Computermania exhibits was the preponderance of games. Businessmen enjoy games, but they wanted to see more business systems up and running; many of the computer systems demonstrated left them with the feeling that all they were good for was game playing.

Computer Stars of the 80s

Will computer programmers be the stars of the 1980s? Think about it and see if you don't agree that this might come about.

If you see, as I do, a strong parallel between high-fidelity and computers, then follow along with me. A hi-fi system does nothing in and of itself . . . you have to have programs to play through it . . . from radio, television sound, records, tapes, etc. A lively business has grown selling high-fidelity hardware . . . and another selling software: records and tapes. How many thousands of hi-fi

Photo 1.

Photo 2.

Photo 3.

Photo 4.

Photo 6.

Photo 5.

stores are there? And, for those of you worried about Macy's and the discount stores selling computers, how many discount stores are selling pseudo hi-fi, the sale of which hasn't seriously affected hi-fi stores?

Computer stores are getting started . . . one in August 1975, 50 in August 1976 and about 500 in August 1977. Will we have 5000 by August 1978? Maybe . . . and maybe even around 50,000 by the 80s. It may be that computer stores will sell the programs . . . or program stores may open, like record stores. We might see as many computer programs as records sold someday – games, educational, business, math, etc.

In the recording industry, top-selling artists make fabulous amounts of money. In the publishing world, top-selling authors also get rich. Why should it be any different for programmers?

Let's look at the money end of things and see where this may be heading. *Kilobaud* is about to come out with some programs on cassettes. The first question is how much should we charge for them? There are factors of how much people will pay, and how much it costs to make, distribute, and sell them. The prices charged for books and records are chosen very carefully . . . as much as people will pay, and as little as will provide a profit for all involved. Computer programs should probably be priced in the same ball park.

How does this work out? With a $7.95 cassette program, the wholesale price to the dealer would be about $4.77 . . . that's a 40 percent discount. Dealers can't provide the stock (not inexpensive), salesmen, premises, demonstration facilities and advertising it takes to move the product with less than 40 percent. That's why this has evolved over the years as a relative standard.

The next man down is a manufacturer's representative. He shows the store salesmen how to sell, checks their inventory, places their orders, makes sure they get delivery, literature, advertising materials, etc. He also keeps track of credit problems for the manufacturer. For all this, he gets 10 percent of the action. The rep is the manufacturer's local salesman.

After taking out the rep's 10 percent, we have about $4.30 left for the manufacturer. If he pays a 20 percent royalty, this brings 86 cents of each sale to the author of the program and leaves $3.44 for the manufacturer's costs and profit . . . which should be enough to make it worthwhile, but not tremendously exciting.

Well, so much for the prices involved . . . what about sales? Where do we get all those superstar fortunes? OK, let's take what

Photo 7.

Photo 8.

Photo 9.

Photo 10.

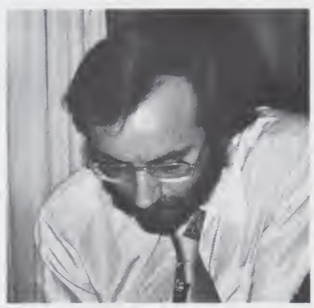


Photo 11.

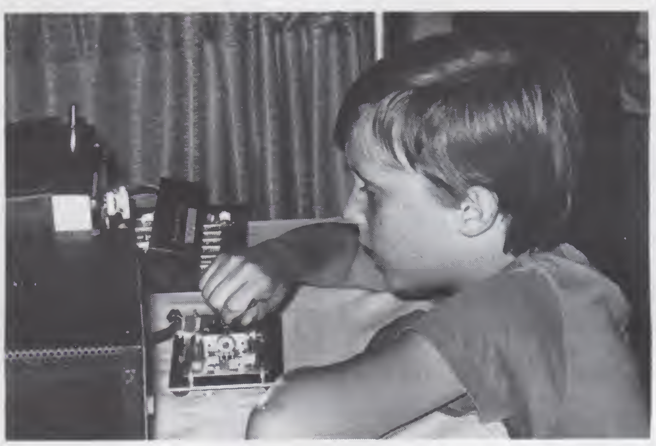


Photo 12.

kilobaud™

Publisher
Wayne Green
Executive Vice President
Sherry Smythe
Editor
John Craig
Managing Editor
John Barry
Editorial Assistants
Steven Fuller
Susan Gross
Production Department
Lynn Panciera-Fraser
Craig Brown
Gayle Cabana
Robert Drew
Michael Murphy
Weston Parker
Bob Sawyer
Noel R. Self
Robin M. Sloan
Typesetting
Barbara J. Latti
Pauline Halvonik
Sharon Horton
Jennifer Johansson
Mary Kinzel
Marie Walz
Photography
Bill Heydolph
Tedd Cluff
Drafting
Bill Morello
Lynn Malo
Associate Editors
Don Alexander
Tim Barry
Sheila Clarke
Rich Didday
Phil Feldman
Tom Rugg
Peter Stark
Bookkeeper
Knud E. M. Keller
Assistant Bookkeepers
Nancy Dupuis
Frances Marion
Jane McQuay
Marketing
Sherry Smythe
Jack Shimek
Cynthia Gray
Circulation
Dorothy Gibson
Barbara Block
Florence Goldman
Rhonda Ramsey
Receptionist
Doni-Anne Jarvis
Computer Data Control
Judy Waterman
Judy Brumaghim
Computer Programming
Ron Cooke
Richard Dykema
Steven Lionel
Service Consultants
Microcomputers Inc.
Nashua NH
Printing
Michael Potter
William Cering
Dwight Perry
Mailroom
Theresa Toussaint
Sue Chandler
Ethan Perry
Maureen Toussaint
Plant Maintenance
Bill Barry
Ken Cox
Lorraine Pickering
Advertising
Mary Jo Sponseller
Heidi Kulish
Marcia Stone
European Distributor
Monika Nedela

Kilobaud is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in the U.S. and Canada are $15 for one year and $36 for three years. In Europe: Kilobaud erscheint montatlich bei Fachzeitschriftenvertrieb Monika Nedela, 7778 Markdorf, Markstr. 3. Abonnement DM 70.- plus Porto 7.20 + gesetzl. MWST; Sfr. 81.- plus Porto 7.20. Please write for other foreign rates. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Publication No. 346690. Phone: 603-924-3873. Entire contents copyright 1977 by 1001001, Inc. INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

we have today and never mind what it is going to be like when we have 50,000 stores. Let's say we are selling programs through only 300 computer stores. A good program might sell how many? Would you go for ten a week? That's 3000 a week for the country (never mind European sales), or 113,000 cassettes a month. At 86 cents each, your programmer is going to be making $11,180 per month. That's $134,000 a year (before taxes).

Now, at $134,000 we're only talking about sales through 300 stores . . . not 3000 (which we should have by next summer). Does $1.3 million impress you? I would like to predict here and now that the time is coming when gold cassettes (or will it be gold ROMs?) will be awarded for million-program sales.

Sure, more involved programs will sell for a lot more, just as specialized books sell for higher prices.

I hope the above is enough to get you started in programming. You learn to be a good programmer by doing it . . . and, with virtually no programs yet being distributed, the field is wide open for just about anything that will sell. What have you got for programs that work . . . will sell . . . and are fully debugged? You could do worse than get in touch with *Kilobaud*.

Mouse Guards Cheese

The only really negative press on Computermania came in one of the "professional" computer publications in an article written by a chap who is selling used computer equipment . . . big used computer equipment.

A few of the computer publications seem to have become used to the idea of microcomputers, but some of the others are exceedingly hostile. It probably has a lot to do with advertising dollars, few of which are coming from the micro industry into the big-computer publications.

Microcomputers are beginning to raise hell with the prices of used computers – and justifiably so. A firm wanting to expand has the choice of buying three-to-six-year-old big hardware or else going with a microcomputer system; and in just about every case, the micro is cheaper and will do more. If you had a warehouse with several million dollars in used big stuff, how would you feel about a show like Computermania that was aimed at the businessman to let him know about microcomputers?

It is unfortunate that so little of the big hardware is compatible with microcomputers. In the early days of micros, I had hoped that we would be seeing articles on the interfacing of this and that piece of hardware . . . so we could take advantage of some of the bargains in used equipment such as tape drives, big disk systems, etc. It turns out that by the time you get through trying to adapt both hardware and software to our new computers, it is seldom worth the fight. Sure, you can buy a used big disk system for a bargain . . . but then comes the need to write an operating system for it . . . and the monthly cost of service . . . etc. Bummer.

Any arguments?

System Costs

There has been growing criticism in the "professional computer" press of the low prices claimed for microcomputers. Perhaps there is some truth to this charge that should be acknowledged.

It is fun to generate enthusiasm among newcomers – and professionals, too, for that matter – with promises of $600 computer systems. The early price of the Altair 8800 kit was an enormous magnet and attracted many computer hobbyists. But once that CPU with its small power supply and 256 bytes of memory had been bought and assembled, the truth gradually began to dawn . . . a little more was needed before this would make a computer . . . maybe a thousand dollars more.

We still need technical articles to explain esoteric concepts such as memory mapping (perhaps a better explanation than has appeared recently in other magazines), and comparisons of the many versions of BASIC. Those of you who have evolved complete systems should pass along your experiences to newcomers who want to know what they can do with a specific combination of hardware . . . and how much it costs. We need articles on low-cost systems . . . medium-cost systems . . . average business systems (which I feel should be in the $12,000 to $15,000 range), etc.

Asking what we want to do with a computer system is begging the question. We want to be able to do anything we can think of . . . write letters, compose music, create art, run a small business, compile mailing lists, etc. We want to interface the telephone to send letters over phone lines. We want to control the temperature of our homes, and make them secure against fire and theft. We want to play games. We also want to be able to do thousands of things that haven't even been thought of yet.

The space available in *Kilobaud* for publishing articles is only limited by the number of pages of advertising. If you patronize our advertisers and make sure they know that you see and appreciate their ads in *Kilobaud*, you'll have a fat magazine . . . packed with as many useful articles as we can get you to write.

Those Fairchild Games

A call from Jim Avery, one of our more interested F8 investigators, revealed that Fairchild seems to be taking a firm stand against hobbyists: little things like a policy of giving no information on their game system, no replacement parts, etc. Despite all this, Jim has delved deeply into the system and finds that it will work quite easily as a CPU. If you are interested in similar pioneering work you might get in touch with him at RFD 2, Box 13A, Leesville SC 29070.

How Much is a Computer?

Many newcomers get as confused as the rest of us did a couple years ago when Mits came out with their first "computer." Sure, it's only a CPU, a box, power supply and a tad of memory . . . but it's a computer.

Let's lay it on the line, readers, and send some letters about your own experiences. What are your requirements for a good working system and what have you in mind to make it even better?

What systems would you suggest for the hobbyist? And how about for the small business? Where would you recommend the newcomer get programs? You've been through the mill, Bill . . . so level with the beginners and let them know what to expect. Tell them what they can do with a given configuration, and how much it costs.

What is the minimum amount of memory that will suffice? How about 24K or 32K? Is anything less than 64K optimum?

There are a few hobbyists out there who have managed to get their computer systems to do more than play games. Let's hear more about what you are doing with your system . . . what I/O boards you need to do it . . . and where you get the programs.

More and more businessmen are buying microcomputer systems, and I assume that a few of them are actually being put to use. Let's have some articles to help the people who haven't been through those woods

(continued on page 20)

EDITOR'S REMARKS

John Craig

Our new business-applications feature starts this month. You may have noticed on the front cover that Bob Brehm's article is "flagged" with a $ logo. In the future this symbol, which is near and dear to us all, will identify business applications articles in *Kilobaud*. These articles will describe small business systems *in layman's terms*. That last point is important.

We, and most of the other small-systems magazines, have published a multitude of material written by system developers for the benefit of other system developers; but very little has been directed to the end user. We're going to feature articles written by system designers for potential customers, and we're most definitely going to have articles written by users of those systems. Spread the word.

DataSync . . . the final chapter

Last month when I mentioned that "Colonel Winthrop" had begun his extended leave of absence (a 2½-year prison term), I thought that was the end of my comments on DataSync.

I'm afraid the company didn't make it. I reported in a previous editorial that they demonstrated their 16K memory board for me . . . and it was working fine. I guess I shouldn't have been so quick to run to my typewriter.

It turned out that the board developed loading problems when it was running in a full system. They pulled their hair out and went without sleep and food for about a month trying to fix the problem. In final analysis, it was a layout problem . . . and by that time they were left with only one choice: close the doors and give up.

Vandenberg Data Products (PO Box 2507, Santa Maria CA 93454) will be offering their 16K static board to DataSync customers at a new, reduced price ($330 kit or $360 assembled). I hope that by now most of DataSync's customers will have heard from VDP . . . and have their 16K of memory.

One final note: The guys at DataSync gave it their best shot. They tried their best to get that product developed and out the door. Those of you who shelled out your hard-earned bucks, and lost most of them, have only one person to blame for this entire mess: "Colonel Winthrop." He taught us all a lot.

As a matter of fact, he taught many of us to cast a critical eye in many directions. Phillip Tubb recently sent me a photograph, which has been in several ads, of a certain 8085 board that has a 16-pin IC (SN74LS) sitting in a 14-pin socket! Looks rather awkward, to say the least. I wonder how many have been delivered that way . . .

Tax Returns . . . on Your Floppy Disk

Tom Scott recently sent me a clipping from *Computer Decisions* that could have a far-reaching impact on home and small-business computer applications. The IRS, in an attempt to cut down on the mountains of paper it has to move, has decided to begin accepting tax returns on floppy diskettes. This new nationwide program will involve sending in machine-readable format produced on IBM System 3, 32 and 34 systems. Initially, the diskettes will contain information for Form 1099 (an information return), and in the future, data for Form 914 (FICA) will be accepted.

Max Stringer, magnetic-tape coordinator for the IRS's Southwest Region in Austin TX, will be directing the program and is now accepting applications for setting up the system.

This decision could eventually have an effect on tax-return filing by individuals, and most certainly small businesses using microcomputer systems. How is the data formatted? How much software would be required to emulate a System 3 floppy disk for a soft-sectored personal system? Sounds like good material for an article.

AROUND THE INDUSTRY

John Craig

I've recently come across some new products (and people) during my travels to companies and clubs . . . and I thought you'd like to meet them.

Godbout's Latest Entry

Look out Heath! Here comes Bill Godbout Electronics with the first of many "foreign attachments" for the H8 system. Although Mits would never go around advertising the fact, perhaps one reason the Altair 8800a and 8800b sell so well is because a wide variety of boards can be purchased for the system. Maybe the same situation will develop with the H8 system if many others follow in Bill Godbout's footsteps.

Bill turned one of his designers, Spencer Cottam, loose with an H8 system (probably one of the first off the assembly line) and had him design a 12K static memory board (that's Spence in Photo 1 holding one of the original boards). The boards were released for production on October 13th and should be ready for shipping by the end of the year.

This 12K board is addressable on any 4K boundary, is static and will be selling for an unbelievable $235. H8 owners will be happy to know they also have an opto-isolator board (has anyone seen Bob Mullen lately?) and an extender board in the works.

Photo 1.

Whew! Maybe Heath won't have to worry about anyone else . . . just Godbout!

The MECA Mass Storage/Audio System

One of the more impressive demonstrations I've seen recently was put on by MECA president Derryl Millican at a meeting of the Micro-8 Computer Club.

Derryl's company has developed a mass storage system (built around two Phi-decks) with a voice-response capability that opens the door to a number of exciting applications.

The Phi-decks are dual-channel, with one channel containing data, and the other voice. Numbers, phrases, sentences or an entire speech can be accessed quickly under computer control. Derryl brought along several demo programs, one of which was a spelling tutor. Karin Ford (John Ford's daughter) was invited to come up and try the program . . . and later had to be bodily removed because she was so taken with it (just kidding). That's Derryl in Photo 2, along with Karin, and he has every right to the pleased look on his face. Karin has been around her dad's computer and a multitude of education programs for several years, but she really is involved in this new experience of the talking computer. In the spelling tutor program, a sentence is spoken and the student is asked to spell a word that was used in the sentence. The little box at the right of the keyboard is a speaker enclosure, and the Alpha-1 tape system is shown on top of Derryl's Altair.

As you may have guessed, I'm enthusiastic about the audio capabilities of the system. I think educational applications are some of the best uses for such a system, but talking games and even voice responses for a business system are other possibilities. (By the way, if anyone ever hears of one of these being used in a "junk-phone-call" system, don't tell me about it. Just wait . . . if those dingalings keep making and selling those systems, we're going to have legislation coming down that may affect all of us!)

The audio is really secondary for most people who purchase the system. It is, first and foremost, a top-notch mass storage system with a sophisticated 3K operating system (with source code!). Give 'em a call or write for additional information: MECA, 7344 Wamego Trail, Yucca Valley CA 92284, (714) 365-7686.

Photo 2.

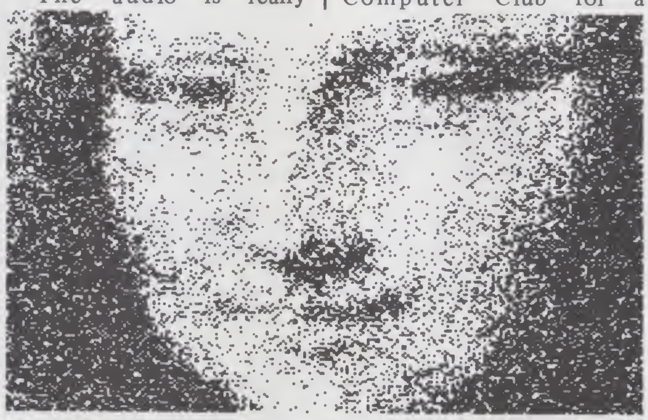


Fig. 1. Graphics output from Malibu Design Group printer.

Your Money's Worth

There are a lot of hobbyists who are actively designing small business, laboratory, educational and other micro-based systems. Without a doubt, one of the most expensive (and troublesome) items these designers have to consider is the hard-copy unit they're going to include in the system.

Mike Sherick recently brought a Malibu Design Group printer (which he designed) to the Santa Barbara Computer Club for a demonstration (Photo 3). Most of us had to be content with drooling, since the price of the unit is not exactly in the hobbyist range – it sells for $1995 (with a rather large discount for OEMs and dealers).

I'm a firm believer in the old saying that you get what you pay for. This thing just reeks of quality – from the walnut-veneer cabinet to the use of a nine-pin Hydra™ print head. It's a dot-matrix printer that zips along at 165 characters per second, and, as you can see from Fig. 1, it has some pretty fancy graphics capabilities, too! (When is someone going to write an article for *Kilobaud* describing a computer-portrait system?)

If you shop around you'll find the price is certainly competitive with other units with the same features and print quality . . . and that Hydra head is top-notch when it comes to dependable service. Forgive me if this sounds like an ad . . . I just liked it! (Malibu Design Group, Inc., 21110 Nordhoff St., Chatsworth CA 91311.)

The PolyMorphic 8813

At that same meeting of the Santa Barbara Club, PolyMorphic Systems released Bob Martin from his cave to tell the folks about his latest software effort,

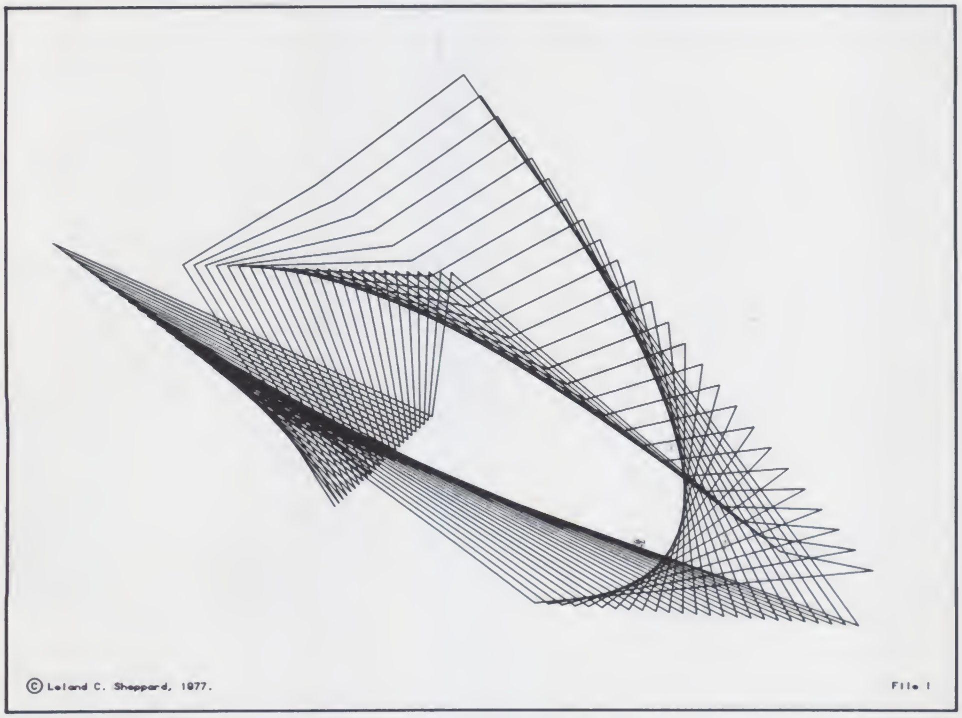
(continued on page 20)

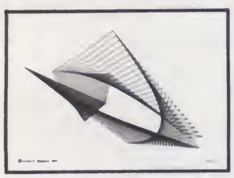
Photo 3.

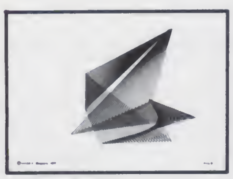
Photo 4.

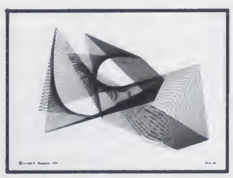
Art-by-Computer$_{tm}$ - an unusual holiday gift idea

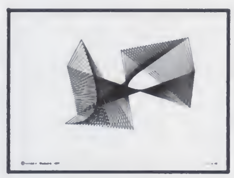
File 1

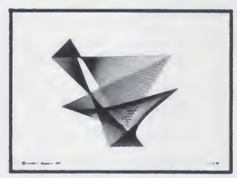


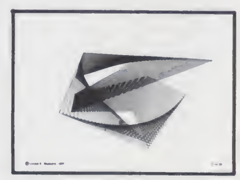

File 1

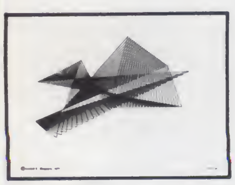

File 3

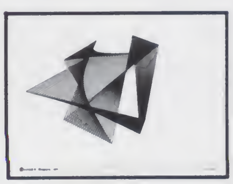

File 13

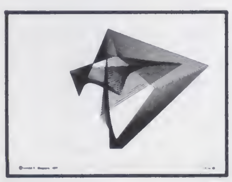

File 19

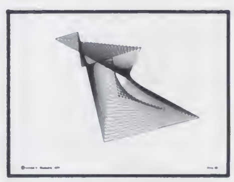

File 20

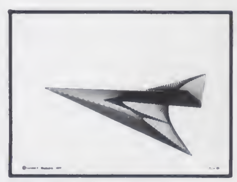

File 26

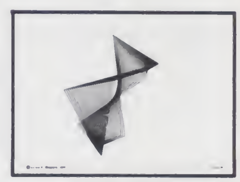

File 27


File 41


File 45


File 46


File 60


File 61

Reproductions of graphic designs generated on a Tektronix 4051 Graphics Computer/4662 Digital Plotter these black-on-white prints are 9 3/4H by 12 3/4W and are suitable for framing or mounting. Price: $2.00 ea./$20.00 for the set of 12, postage paid. Specify File #s or Set #1. Send check or money order to: Leland C. Sheppard, Dept K, PO Box 60051, Sunnyvale, California 94086. California residents add 6% sales tax. Allow 2 weeks for personal checks to clear.

NEW PRODUCTS

Video Display Board from TDL

Technical Design Labs has introduced the VDB (Video Display Board), a video interface for the Altair bus microcomputers.

The unit consists of two boards, one piggybacked to the other. The unit occupies one edge connector on the bus, but takes up the space of two boards.

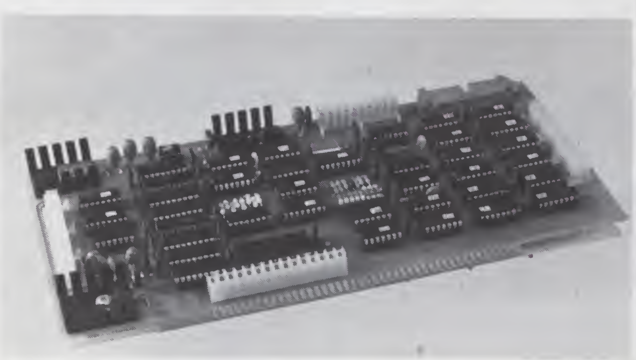
The VDB contains its own display buffer memory and provides two pages of display, each with 25 rows of 80 characters. The display buffer memory does not use any memory address.

This new product displays, in addition to the 96 uppercase and lowercase ASCII characters with decenders, 64 unique display symbols, permitting a graphic resolution with 160 horizontal elements by 75 vertical elements. The display can accept data at a 400,000-character-per-second rate.

The VDB provides a true hardware blinking cursor to facilitate the programming of special edit functions. It is addressable and indicates the physical location on the display screen where the next symbol will be written into or read from.

The VDB works with either a modified TV set or monitor and has an on-board 8-bit parallel keyboard port with status strobes. The VDB requires one motherboard socket and occupies two card spaces.

The VDB is priced at $349 in kit form and $449 assembled and tested. Technical Design Labs, Research Park, Building H, 1101 State Road, Princeton NJ 08540.



Technical Design Labs' VDB.

TSC Multi-User System

Technical Systems Consultants' Multi-User System allows four users to simultaneously operate one SWTP 6800 microcomputer, all running separate programs.

The TSC Multi-User Board is a SS-50 bus board containing some required extra memory, interrupt logic, and a few registers. The board is designed to plug right into one of the memory slots on the bus. No machine modifications are required. With the board installed, simply load up the BASIC cassette included with the board and you have a four-user BASIC system.

Suggested retail price for the TSC Multi-User Board Kit is $129.95. That includes the Multi-User Board Kit with all parts, IC sockets, diagnostics, and instructions. The boards are high quality with plated-through holes. Also included in the price is a cassette and users manual for a Four User Micro BASIC Plus.

Technical Systems Consultants, Inc., P.O. Box 2574, West Lafayette IN 47906.

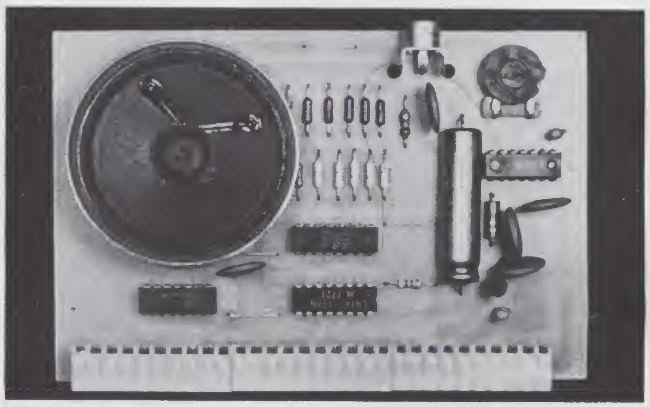
Music for SWTP 6800 Owners

The Newtech Model 68 Music Board enables the user to generate music, sound effects, rhythms, Morse code, and touch-tone synthesis.

The Model 68 Music Board, designed for the SWTP 6800 computer, comes fully assembled and tested. It consists of a digital-to-analog converter, audio amplifier, speaker, volume control and phono jack for convenient connection to an external speaker or home audio system.

A complete user's manual includes sound effect programs, test routines, a BASIC program for writing musical scores and a 6800 assembly language routine for playing them. An AC-30-compatible cassette contains programs from the user's manual and software for pre-coded songs. The Model 68 Music Board is $59.95.

Newtech Computer Systems, Inc., 131 Joralemon Street, Brooklyn NY 11201.



Newtech Model 68 Music Board.

CP&L Offers Word Processor

Computer Power & Light, Inc., has a commercial quality, microcomputer-based word processing system for under $6000, complete. Based on the field-proven COMPAL-80 computer and Xerox Corp.'s Diablo 1620 daisy-wheel printer, it con-

CP&L's word processor.

tains features found on systems costing $20,000 or more. Among these are complete text editing on a large CRT; insertion or deletion of text and the ability to move blocks of text anywhere; variable-speed scrolling of entire text on the CRT, forward and backward; ability to search for all occurrences of a specific word or group of words and replacement with alternative word or words; and storage and retrieval of finished text on low-cost Phillips audio cassettes at the rate of 240 characters per second.

There is also a variety of printing options, including variable line length, 1-5 spaces between lines, variable character spacing, presettable page headings, page numbering, and right and left margin justification using the Diablo's unique character spacing routines. No extra blanks are inserted in your text, nor is there any need for hyphenation.

Computer Power & Light, 12321 Ventura Blvd., Studio City CA 91604.

Dynabyte Static RAM

Two fully static RAM modules for the Altair bus are now available from Dynabyte, Inc.

The 16K and 32K static RAMs are available with access times of either 450 or 250 ns. The 250 ns RAMs are compatible with 4MHz Z-80 processors.

Like Dynabyte's 16K dynamic RAM module, both static RAMs are completely assembled, tested, burned in and guaranteed for one year.

The new RAM modules' fully static functioning and complete buffering make them compatible with all known processors, including the Z-80s offered by several manufacturers, as well as the Alpha Microsystem A-100.

Both 16K RAM modules feature Bank Select, which allows up to eight separate banks (of up to 64K each) to reside in the same system. The module may be addressed in four separate 4K blocks along 4K boundaries. Each of these blocks may be individually write protected. If an attempt is made to write into a protected block, an audible alarm will be activated and a visual indicator will be displayed for several seconds.

The 32K static RAM modules offer 4K boundary addressing, complete buffering, and conservative thermal design.

The modules employ Schmitt trigger buffers on inputs and Tri-state TTL buffers on outputs.

Dynabyte, Inc., 4020 Fabian, Palo Alto CA 94303.

North Star Executive Software

XEK, a complete system executive package for North Star users, is now available from the Byte Shop of Westminister CA.

The XEK package contains a disassembler capable of creating files that may be left in memory when changing from the disassembler to the executive package for reassembly. The monitor software can accept input from cassette tapes and paper tapes as either source or object files, as well as from the North Star diskette system.

The assembler also features a new auto-line editor for the creation of source files. This editor extends to the modification of existing object files.

Another feature is the XEK's ability to handle up to six named files at once that may be consecutively assembled to form one object file. The assembler, monitor, and disassembler come with complete documentation, both on disk and as a manual. Price, including California sales tax, is $48.

The Byte Shop of Westminister, 14300 Beach Boulevard, Westminster CA 92683.

Static Memory Modules For PCM-12

Pacific Cyber/Metrix, Inc., has three static memory modules for use with its PCM-12 – the firm's 6100 microprocessor-based, 12-bit microcomputer system. Each module is available either fully assembled or in kit form.

The PCM-12 system is fully software-compatible with Digital Equipment Corporation's PDP-8 family of minicomputers.

The most basic system memory module, designated the 12020A, is a 4096-word by 12-bit memory element that constitutes one full field of memory. The PCM-12 can accommodate up to eight of these boards, which give the system a 32K memory capacity. The 12020A costs $289 assembled and tested ($199 kit).

A second memory module is a combination of EPROM and RAM devices. With PC/M's 12040A Memory Extender module, the 12160 can be used in any 4K field of memory in the PCM-12 microcomputer system. The single-quantity price of the 12160 is $455 ($385 kit).

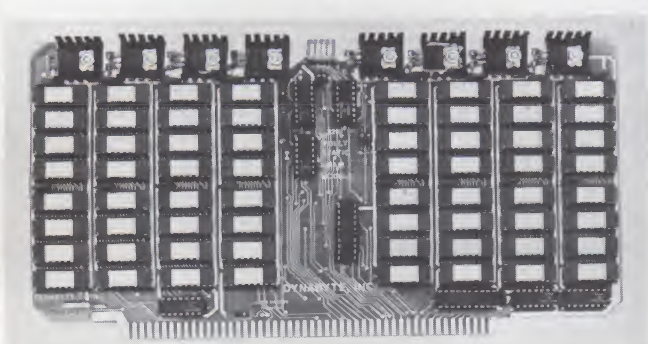
The third module, the 12210, is a non-volatile memory organized as 4096 12-bit words. With automatic-recharge battery back-up, the 12210 module will retain the contents of its memory for up to thirty days with system power turned off. The 12210 4K CMOS memory module is priced at $580 ($490 kit).

PC/M, 3120 Crow Canyon Road, San Ramon CA 94583.

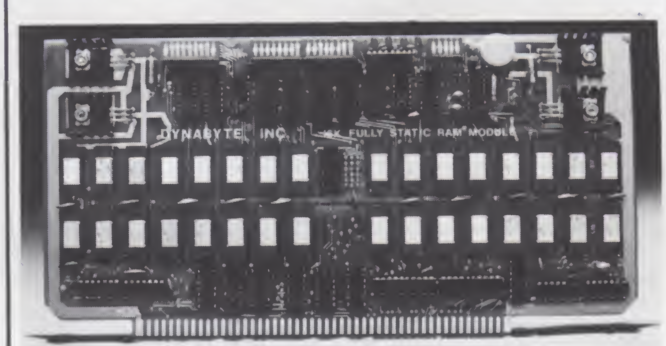
Upgrade Kit For Poly 88

PolyMorphic Systems, Inc., has introduced an upgrade kit for Poly 88 microcomputer owners who wish to convert their current systems to the company's new System 8813 disk-based microcomputer system.

The Poly 88 Disk Kit contains all mechanical parts and electronic assemblies needed for converting a Poly 88, including chassis, walnut cabinet, a ten-slot backplane, power supply, floppy disk controller, 2K of read-only memory (ROM), a fan, one floppy disk drive and two system



Dynabyte 32K Static RAM.



Dynabyte 16K Static RAM.

diskettes.

The conversion to a disk system takes only a few hours. The Poly 88 Disk Kit is priced at $1450, and up to two more disk drives may be added at $590 each.

PolyMorphic Systems, Inc., 10832 Ventura Blvd., North Hollywood CA 91604.

Double Density Floppy Disk System

Recording density is hardware and/or software selectable on the new FDC-3 controller board from Digital Systems. The user may choose either IBM 3740 format or a double-density format of 571K bytes per diskette (77 tracks of 58 sectors with 128 bytes per sector).

The new, lower-power controller board is available separately or in 8080 systems consisting of up to eight Shugart Associates Disk Drives and operates with CP/M software, a complete disk operating system including a text editor, assembler, and dynamic debugging tool. A write protect option can be hardware enabled or selected with notched diskettes.

Other features include DMA, hardware bootstrap loader, CRC error check, and Z-80 compatibility. An Altair interface card is included. Interface cards are also available for other 8080 bus systems. A fully assembled and tested dual drive system, complete with cabinet, power supply and all cables, is $2545 for single density or $2745 for dual density.

Digital Systems, 6017 Margarido Drive, Oakland CA 94618.

Timesharing Microcomputer

MicroAge, the systems marketing division of the Byte Shops of Arizona retail computer stores, has a free eight-page brochure on the Alpha Micro multiuser multitasking timeshared microcomputer system.

The Alpha Micro is a powerful Altair bus compatible software development system that features a timeshared operating system with full utilities, multiuser structured file system with password security, disk file management system for floppy or hard disks, AlphaBasic extended compiler and reentrant run-time software, free-form text editor and text formatter, multiple pass Macro Assembler and 16-bit microprocessor with hardware floating-point computation.

Complete systems can be configured with multiple terminals and peripherals, including hard-disk systems with up to 1200 megabytes on-line. Disk access methods incude sequential, random and indexed sequential. Other languages currently under development are LISP, APL, FORTH, 8080 Cross Assembler, with FORTRAN, COBOL and RPG languages in the planning stages.

The Alpha Micro is based on Western Digital's advanced WD-16 microprocessor chip set (the same set used in the PDP-11 series) with three sister microprogrammed ROM chips for a highly flexible macro instruction set. The 16-bit processor interfaces to the popular 8-bit Altair bus by multiplexing through the CPU logic board. This multiplexing is totally transparent to the programmer.

MicroAge, 803 N. Scottsdale Road, Tempe AZ 85281.

Million-Byte MetaFloppy System

Micropolis Corporation has expanded its field-proven Disk Extended BASIC software package to provide additional support for BASIC programming with its new million-byte Model 1054 MetaFloppy system.

The expanded BASIC includes a flexible new CHAIN command, which allows the user to segment very large programs and run the segments in any order. Thus, the new command permits running of programs which are larger than the computer's memory by using the disk as intermediate storage.

Another new feature, of particular interest to business-oriented users, is a powerful line-printer support capability. This not only lets users print output directly to a line printer but also to store it on the disk for later printing or examination if desired.

Standard business-oriented features include variable precision arithmetic, complete STRING and substring capability, and extensive disk file commands.

The new Micropolis BASIC is designed for 8080/Z-80-based microcomputers having at least 24K bytes of random access memory (RAM).

Micropolis Corp., 7959 Deering Ave., Canoga Park CA 91304.

8080 Word Processing System

Mini Word Processing 2.0 (MWP) enables the user to prepare letters, text and mailing labels or envelopes. When used for correspondence processing, MWP allows name and address entries to be coded with a number of group codes and document response codes. For example, an inquiry might be coded with date and inquiry-type group codes and a specific response letter body with selected paragraph/phrase insertion document codes. A follow-up letter might be sent keyed only on group codes.

MWP provides in-line editing and common text/phrase insertions in the text generation module. The letter and text output modules provide text insert or replacement, margin/page control, as well as page numbering.

Double Density Floppy Disk System.

Micropolis Model 1054 Metafloppy system.

MWP is extremely easy to use and includes a comprehensive user's manual with varied examples. The price is $195 supplied on a diskette compatible with Mits Disk Extended BASIC. The Software Store, 706 Chippewa Square, Marquette MI 49855.

ANSI-Standard FORTRAN IV

Technical Design Labs has ANSI-standard FORTRAN IV available for microcomputers, written for them by Small Systems Services, Inc.

FORTRAN IV is the first complete ANSI-standard FORTRAN IV to be offered on the market, and is equipped with many extensions that exceed ANSI standard requirements.

In addition to compiler, loader, libraries, DOS linkages, *etc.*, this FORTRAN IV package may be interfaced to TDL's forthcoming Hardware Multiply/Divide module. It will then run 10 to 50 times faster than the fastest BASIC packages available.

Operationally, this version is a disk-oriented system. It runs in less than 24K with DOS, and both FDOS IV and CP/M versions are available.

The complete package includes both the floppy diskette with object code and a user's manual. Additional documentation and support packages are available. It is priced at $349.

Technical Design Labs, Inc., Research Park, Building H, 1101 State Road, Princeton NJ 08540.

Extender Board with Built-in Logic Probe

Compatible with the Altair bus configuration used by over a dozen manufacturers, this board allows easy troubleshooting and examination of computer boards. The built-in logic probe indicates low-level logic (green LED), high-level logic (red LED), and high/low transition or pulse (yellow LED remains lit for about .2 second to catch short pulses).

A specially designed edge connector allows use of clip-lead probing; the edge connector's label gives pin numbers, and locates power and ground connections.

Jumper links in the power lines (+8V, +16V, -16V) allow easy current measurement and fusing of the board under test. By adding a switch in these lines you may shut down board power without turning off your computer.

The Mullen Extender Board is available in kit form for $35. Mullen Computer Boards, PO Box 6214, Hayward CA 94545.

8K ROMs Rated at 55ns

Monolithic Memories, Inc., has a series of 1024-by-8-bit bipolar ROMs rated at 55 ns T_{AA} and prices at $10 in volume for commercial grades. The new 8K ROMs are also available in MIL-temperature range rated at 70 ns access time.

Typical delivery is five weeks from receipt of customer's bit pattern; first-time masking charge is $750 per pattern in minimum quantities of 100.

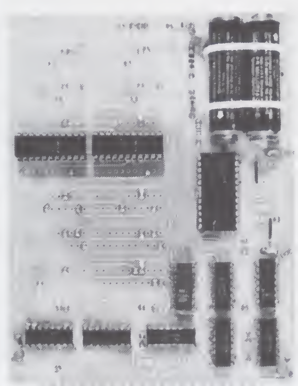
Monolithic's 8K ROM family comes in a 20-pin package (6289-2), a 22-pin version (6286-2) and a 24-pin package (6280-2) – all designed as drop-in replacements for other vendors' memories. The 22- and 24-pin DIPs are also compatible with Monolithic's off-the-shelf bipolar PROMs.

Monolithic Memories, 1165 East Arques Ave., Sunnyvale CA 94086.

6502 Card for Altair Bus

CGRS Microtech has a new 6502 microcomputer card for the Altair bus structure. With the CGRS Microputer 6000, Altair bus users can take advantage of a standard packaging scheme and still enjoy the high speed and versatility of the 6502. The CGRS Microputer 6000 computer card contains 4K of EPROM (2708) and 2K of RAM (2111) in addition to the 6502 microprocessor and TTL support logic.

CGRS Microtech, PO Box 368, Southampton PA 18966.



CMOS RAM by Wintek.

Nonvolatile CMOS RAM

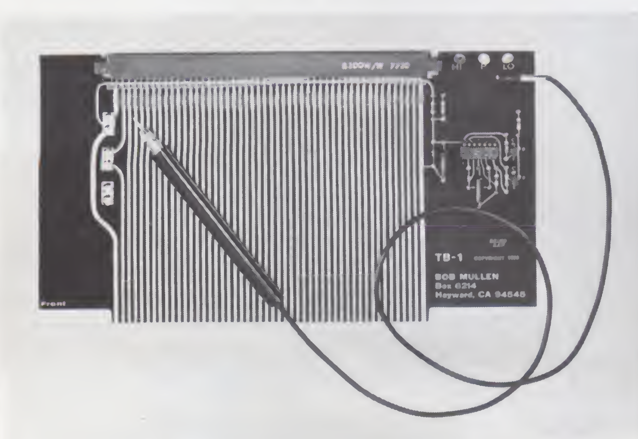
Microprocessor memory expansion with nonvolatile memory is now possible with the Wince CMOS RAM/Battery Module. Memory is retained during power-off conditions including when the module is unplugged from the system bus. Two size AA nickel cadmium batteries allow for power-off periods up to one year. The module can accommodate up to 2K bytes in multiples of 256 and has write protection. The price is $399. The Wince Micro Modules are unique in that they are the only 6800-based uP modules available on industry-standard 4½ x 6½ inch 44-pin printed circuit boards.

Wintek Corp., 902 N. 9th Street, Lafayette IN 46904.

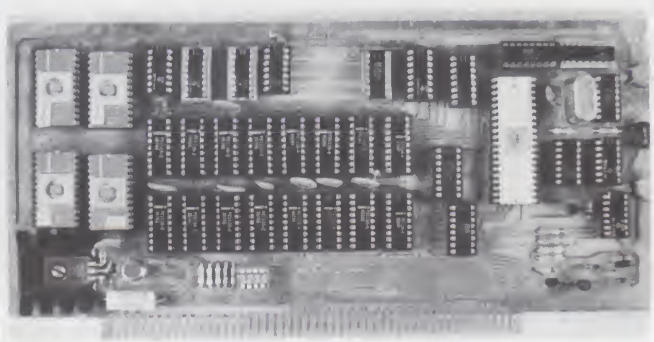
Business Software Book

Microcomputer Business Systems, Inc. announces a

(continued on page 112)



Mullen Extender Board.



CGRS Microputer 6000 Card.

BOOKS

How to Program Microcomputers
William Barden Jr.
Howard W. Sams & Co. Inc.
$8.95.

Programming in machine/assembly language using the microprocessor instruction set is often confusing for newcomers. This book by William Barden Jr. ranks as one of the best I've found to help minimize the confusion. In addition to sections on computer technology and machine architecture, the book provides a thorough comparison of programming techniques used for the three most common microprocessors (8080, 6800 and 6502). For this reason alone, it rates a place on your bookshelf.

Part 1 reviews the recent history of microcomputers and describes the binary number system, arithmetic operations, operations of the microcomputer, and microcomputer data codes. This is a section of the book that I think should be at the end; possibly as an appendix. If you're already familiar with these subjects, skip these pages and get to the real meat; if not, you won't need them to read the book. The information is only for reference and review, and is not extensive enough to be tutorial.

In Part 2, the register architecture and descriptions of the addressing modes are provided first. A block memory map of the 8080, 6800 and 6502 systems accompanies the next chapter describing memory and stack operation. Chapter 9 has the best description and comparison I've seen on the grouping, by category, of microprocessor instruction sets. The remainder of Part 2 describes the input/output and interrupt processing techniques used to talk to peripherals.

"Assembly Language Programming with Microcomputers" is the title of Part 3 and, I think, the best part of the book. A thorough description of assembly language programming is provided, and after reading it, I was able to convert 8080 and 6800 programs for use with my 6502 system. Of course, this works in the other directions too.

In "Programming Algorithms" (Part 4), Mr. Barden provides a number of routines for each of the microprocessors. There are over 18 programs ranging from timing loops to random number generators. These programs can be used by themselves or as subroutines in larger programs. Again, having routines in all three instruction sets for comparison provides a way for converting programs from one instruction set to another.

An appendix of the instruction sets for each of the microprocessors is included at the end of the book. What more could you ask for in an all-in-one-place assembly language programming book for your computer library?

Chuck Carpenter
Carrollton TX

Digital Experiments
Richard E. Gasperini
Movonics Company
Los Altos CA 94022, $8.95

Students of "Kilobaud Klassroom," rejoice! *Digital Experiments* is the perfect supplement to the monthly experiments presented by George Young in the pages of *Kilobaud*. Furthermore, using the "Kilobaud Klassroom" Student Console to perform the experiments would be a lot easier on the pocketbook than using the design console the author more or less recommends (the Hewlett-Packard 5035T Logic Lab, which sells for $750!).

The book is about evenly divided between the digital experiments and four appendices, which contain IC data sheets, small-scale and medium-scale ICs (Appendix A and B), an IC cross-reference and a list of IC suppliers. The appendices are reprints from the Texas Instruments *TTL Data Book* and cover all of the ICs used in the experiments. The idea of including them in the book, rather than forcing the student to buy or have access to one, is a good idea. Another nice feature is the index (a worthwhile feature in any book).

The experiments really can (and should) be set up on the "Kilobaud Klassroom" Student Console. The only modification to the console would be the addition of six switches. The specifications for the logic lab are as follows: must have a breadboard for 6 ICs; a 5-volt, 1 Amp power supply; six to eight data switches; four to six level indicators; a slow (1 Hz) and a fast (greater than 1 kHz) clock.

The book's objective is to provide a hands-on approach for the digital-electronics student, and after going through several of the experiments I heartily recommend it. The experiments range from the simple inverter and lighting an LED, to multiplexers, displays and RAM and ROM memory chips. In between, you will find digital experiments one would expect in a book such as this, including those dealing with the four basic logic gates, flip-flops, counters and decoders.

Aside from the logic lab discussed earlier, the book also requires that you have a logic probe (built into the Klassroom console) and a logic clip (which snaps over an IC and displays its input and output levels on an array of LEDs).

John Craig
Lompoc CA

An Introduction to BASIC
M. R. Eagle
G. Bell and Sons Ltd., 1976

This delightful book from England (USA Distributor – Transatlantic Arts, Inc., Levittown NY 11756) should be in the hands of every serious beginning BASIC programmer. Written in an easy-reading, no-nonsense style, an amazing amount of material is packed into 90 pages containing seven chapters. There is also an appendix, 38 pages of solutions to the exercises and an index of key words and phrases.

Regardless of how many books your computer library contains, there should be room for one from an out-of-the-country source. It provides a fresh viewpoint and a variety of terminology and approaches to the fascinating field of computers.

After a four-page introduction, the author jumps right into the language. However, these four pages offer some insight into the use of BASIC in England. The "computer store" referred to has nothing to do with your friendly neighborhood retail computer outlet. It refers instead to the computer's memory (RAM). Thus, ". . . programs and data are transmitted from the input device to a store. . . . This is called a *fast-access store* because instructions and results can be transmitted to and fro between it and the central processor. . . ."

BASIC interpreters are not mentioned, leading you to believe that the small, personal computer had not made its impact when the book was written (1976). "When you put in a program it is first of all *com-*

piled and the coded version is held in your workspace. Once this is done the whole program can be *executed* at full speed."

The book refers to external memory: "Many computer systems have back-up stores for holding files of programs and data, compilers for different languages and so on. These may be in the form of magnetic discs, drums or tapes. . . ."

On the computer system: "The whole package, program plus commands plus any additional data, is known as a *job*. . . . A common type of computer service . . . runs a *batch* of jobs. . . . Interactive use of the computer . . . this method of input and output is extremely slow, and is usually operated under a time-sharing system with multiple access."

Chapter two introduces the FOR-NEXT loop and the PRINT and END statements. READ, DATA, INPUT, LET, GOTO and IF-THEN are also treated in this chapter, as are strings and relation operators (=, <, > etc.). By the end of the chapter, the beginner is well into BASIC usage.

The third and fourth chapters discuss program planning with flowcharts, algorithms and dry runs. Standard BASIC functions and facilities such as user-defined functions, REMark, RESTORE and ON GOTO statements are introduced.

Iterative methods and loops are more thoroughly explored in chapter five. Graphic techniques are introduced in this mathematics-oriented chapter.

Chapters six and seven explain tables, lists, arrays and their applications while introducing DIM and MAT statements. A discussion of subroutines and the GOSUB and RETURN statements completes the text material.

Appendices include use of a terminal, debugging program errors and a summary of the BASIC language.

According to the preface, "This book is intended as a first introduction to programming using the BASIC language." As a former high-school teacher, I think the book would make an appropriate text for a one-semester introductory course in BASIC programming. For the hobbyist, it presents an organized introduction to BASIC, something that is not always readily available. The exercises provide good examples (although, as usual, largely mathematics-oriented), with complete solutions.

Don Inman
Menlo Park CA

THE HEATHKIT FORUM

Charles Floto

So far, only a Maryland reader has responded to my question in the October *Kilobaud*. Timothy M. Markell reports: "My H8 was up and running with 4K of RAM on September 26, 1977. Total assembly time for me so far has been about 30 hours: 25 hours for the H8 and five hours for the H8-1." I suspect several of you had an H8 going before September 26th, so let's hear about it.

Timothy would like Heath to produce a revised "section of the H8 manual, expanding their discussion of troubleshooting techniques and adding a 'symptom/cause/cure' table for use in conjunction with the troubleshooting charts."

Floridian Paul Kanciruk wants this Forum to "survey the hobby computer industry to determine to what extent they will produce hardware compatible accessories such as memories, interfaces, speech synthesizers, etc., to augment Heath's offerings." Consider it done, Paul. I look forward to reporting the results.

Paul also comments: "The H11 is more expensive than the average home computer, but I believe it represents a step forward. Did Mr. Craig ("Around the Industry," issue No. 9) favor the old four-bit machines over the newer and more expensive eight-bit computers because they were good enough? If there is one constant in this field, it is that things are not static, that ongoing upward evolution/revolution is the rule."

I went to Alexandria VA for the grand opening of a new Heathkit Electronic Center. As soon as I'd entered the door-prize drawing I noticed an H8 just sitting there being almost ignored and with the front-panel digits blanked out. So, I dropped into the chair placed conveniently in front of it and began to explore what the keyboard would do. Specifically, I wanted to vary the length of the tone from the built-in speaker (the frequency is fixed).

Just when I was getting the hang of it, a salesman came over, lamenting that I'd destroyed the BASIC program he'd been running on the H9 video terminal at the other end of the table. Although it didn't take him long to reload the program from cassette, the occurrence did raise a question about the H8. How do you keep people from crashing your system by playing with the keyboard? At least Heath puts the power switch on the rear and requires a person to press two keys at once to do anything drastic. What can you suggest?

Speaking of the front panel: I've received a response to John M. Blalock's question in the October Forum about the need for one. Heath's computer products director Louis E. Frenzel acknowledges: "The question about the front panel on the H8 Digital Computer has come up several times recently. We all know that the H8 could get along fine without a formal front panel, but such a version does not provide the convenience, flexibility and computing power that the front panel provides. You simply cannot appreciate what the front panel does until you have actually used one. It is super!"

He goes on to cite the unique front-panel capabilities: "That's right, you can actually watch the content of the register or memory location changing as it happens. You can't do that with a debug routine on a terminal. Another function that is very helpful is the single-step feature, which cannot be physically accomplished via a video terminal."

These statements about a terminal's limitations may have been made with the H9 in mind. As Leo Taylor has called to my attention, it is designed to be used with the H8 at a maximum speed of 600 baud. That's slower than the maximum rate of 9600 baud advertised for both the H9 and the H8-5 serial interface; but it is what the manual calls for when you put the two together.

Both John Nierste of Michigan and Raymond E. Penley of Maryland expressed disappointment at Heath's use of the 8080 rather than a newer eight-bit processor. I want to point out that nothing about the H8 bus or front panel requires that they be used with the 8080. Heath, or another company, could easily produce a CPU card for the H8 using a different processor and monitor ROM. In fact, I'll be surprised if there isn't an H8 CPU card using the 6800,

(continued on page 21)

LEGAL/BUSINESS FORUM

Kenneth S. Widelitz
Attorney at Law

Last month we looked at implied warranties. As you recall, implied warranties are created by the Uniform Commercial Code (UCC), which exists in almost every state, and by the Song-Beverly Consumer Warranty Act (SBCWA) in California. This month, we will examine express warranties, emphasizing the federally enacted Magnuson-Moss Warranty Act (MMWA) and some of the unique provisions of the SBCWA relating to express warranties.

Remember, express warranties are usually written warranties and for the purposes of the MMWA must be in writing.

"Full" versus "Limited" Warranties

A very important feature of the MMWA is its requirement that *every* written warranty accompanying a product selling for $15 or more be clearly designated either "full" or "limited." If a warranty merely states "warranty" at the top, it fails to meet these federal standards. Rather, the warranty must be designated, for example: "full 90-day warranty." A warranty is designated "full" when it meets the federal minimum standards for warranty set forth in the MMWA.

The requirements necessary in order to obtain a "full" designation are as follows:

1. The warrantor must remedy the defective product within a reasonable time and *without charge*. Thus, a warranty which requires that the consumer pay for transporting the product to a repair facility must be a "limited" warranty, as the remedy is not provided without charge.

The term "remedy," as defined by the Act, means either repair, replacement or refund. It should be noted that the MMWA only requires that the remedy be given within a reasonable time. The SBCWA requires that goods be repaired within 30 days.

2. The written warranty may not impose any limitation on the duration of an implied warranty.
3. The warrantor may not exclude or limit consequential damages for the breach of any written or implied warranty, unless such exclusion or limitation conspicuously appears on the face of the warranty.
4. The warrantor may not impose any duty upon you other than notifying the warrantor of the need for repairs unless the warrantor has demonstrated that such a duty is reasonable. Such a reasonable condition might be the filing of a warranty registration card, if such requirement appears on the face of the warranty. This provision prohibits a manufacturer from giving a "full" warranty on a memory board only if it is used with his CPU. If he wants his warranty to be effective only when the product is used with his other products, he can do so, but then he can only give a "limited" warranty.
5. The warranty must extend to each person who is a consumer under the MMWA. The MMWA defines consumer as a buyer *or any person to whom the product is transferred* during the duration of an implied or written warranty. Thus, the manufacturer giving a "full" warranty cannot deny liability to virtually anyone who obtains possession of a product during the term of the warranty. The SBCWA limits its scope to only the retail buyer of a consumer good.

What a Warranty Must Disclose

Whether a warranty is full or limited, the MMWA requires that on its face it disclose a great deal of information. The warranty must set forth the identity and address of the warrantor and the identity of the parties to whom the warranty is extended. It must state a clear description of the parts, or characteristics, or components covered by the warranty and, where necessary for clarification, which are excluded from the warranty. The warranty must also state what the warrantor will do in the event of a defect, for what period of time and at whose expense. It must also include a step-by-step explanation of the procedure the consumer should follow in order to obtain performance of any warranty obligation. The warrantor must further provide a general description of the legal remedies available to the consumer, and any limitations on incidental or consequential damages.

If a warrantor fails to designate the warranty as either "full" or "limited" or fails to disclose required information, the warrantor is subject to a lawsuit brought by the attorney general or a Federal Trade Commission attorney. A civil penalty of up to $10,000 per violation may be assessed.

As of this writing, conversations with FTC attorneys indicate that, as yet, no public actions have been brought to enforce disclosure requirements. The explanation is that since the rule has been in effect for such a short period, the FTC is allowing additional time for warrantors to comply. However, there have been some citations issued that have not been made public. I have been told that the violating parties are conducting what the FTC designates consent negotiations.

Presale Availability Rule

Effective for products manufactured after January 1, 1977, the Presale Availability Rule will have a vast impact on consumer awareness of warranties and on the manufacturers' and retailers' costs. The rule requires that the seller of a consumer product costing more than $15 make the written warranty accompanying such product available for the prospective buyer's review, prior to sale, by the use of a few alternative methods. This rule is a creation of the MMWA and applies to every retailer and manufacturer in the country.

One method provides for the text of the written warranty to be displayed in close conjunction to each warranted product. Another method allows the retailer to maintain a binder (loose-leaf notebook) containing copies of the warranties for the products sold. The binder must be maintained at a location that provides the prospective buyer with ready access. Such binder must be indexed according to product or warrantor and must be kept up to date when new warranted products or models, or new warranties for existing products, are introduced into the store.

Under this method, the retailer must display such a binder in a manner reasonably calculated to elicit the prospective buyer's attention – or place signs indicating the availability of the binders in prominent locations in the store. A third available method is dis-

(continued on page 21)

LETTERS

From the Ashes — TIPS

As you are probably aware, *The Computer Hobbyist* newsletter is now defunct. This has alarmed many home computer hobbyists who had depended on TCH for some of the best construction articles to be published to date. For many home computer hobbyists, these construction articles were the only alternative to purchasing commercial boards and kits. I see as the only hope now that the major hobby-computer magazines might try to fill the gap left as these newsletters go under.

TCH was in the process of running a series of four articles on constructing an IMP-16 computer system when it ceased publication. Hal Chamberlin designed this system, and from comments about him and TCH in previous issues of *Kilobaud*, I know that you are aware of his capabilities. A friend of mine, Tony Lynch, and I were two of the first people to build this system. Based upon our experiences, this system deserves much more attention than it has received to date. The major problem is that only the first two articles were printed in TCH. The third article was printed in National Semiconductor's newsletter, *Compute*, and the fourth and final article will soon be sent by Hal Chamberlin to National Semiconductor so they may also publish it in *Compute*.

To continue support for this system, Tony Lynch and I have started a TCH IMP-16 Punibus System (TIPS) newsletter.

We are advising you of our plans first as we believe *Kilobaud* is the best magazine for the home-computer hobbyist.

Also, I would like to inform you that the Greenville Area Home Computer Club has been in existence for over a year now. Monthly meetings are held at Greenville Technical College on the second Thursday of each month. The most recent accomplishment of the club is the development of a microprocessor-based (SC/MP) TVT which uses a 5320 sync generator, MCM6570 series character generator and is user-selectable for 32 or 64 character lines.

Frederick R. Holmes
101 Brookbend Court
Mauldin SC 29662

Which One Will Work?

I am a new computerist. Like everyone else, I'm confused . . . learning new words, trying to understand what the spec sheets mean. What is the difference between a 4K RAM board and a 4K RAM board. That is not a misprint. Here is what I'm talking about.

In the September issue of your magazine, on page 122, there is an ad for a 4K RAM board kit — fast, low power 2102-1, (450 ns), fully buffered, standard 44-pin gold-plated connector; complete kit includes board, chips, caps and documentation. In the same issue (p. 118) is an ad for an Imsai 4K 500-ns RAM kit.

The first kit was priced at $79.95; the second kit was priced (with discount) at $117.00. My question is, Why? Please help me. Will the cheaper board work on my KIM-1; and if not, why not? Thank you.

Now that I have that off my chest, I want to tell you what a great magazine I think you have. Kilobaud Klassroom is most valuable to me. I like to build my own equipment, so your construction articles are great. Keep up the good work.

Murray Smith
Rochester NY

From what I've seen, Murray, it appears that the first board you mentioned is a snap to interface to a KIM. At any rate, we have an article coming up in the near future on that subject. If you want to go with the Imsai board and the S-100 bus you should explore John Blankenship's "Expand Your KIM!" series. — John.

Three at One Time

In his article "Time for Timesharing? . . . some hints and pitfalls" (*Kilobaud* No. 10), Ken Knecht indicates that "Mits limits you to writing on only one sequential file." Mr. Knecht is confused. As near as I can tell, the only file limitations are those that are self-imposed during system initilization.

The attached program, written on my Altair 8800b under Disk Extended BASIC version 4.1, shows three files open simultaneously for output. They are then closed and reopened for input to verify that the output actually took place.

Irwin Doliner
Pikesville MD

```
10 OPEN "O",1,"FILE1"
20 OPEN "O",2,"FILE2"
30 OPEN "O",3,"FILE3"
40 PRINT #1,1
50 PRINT #2,2
60 PRINT #,3
70 CLOSE
80 OPEN "I",1,"FILE1"
90 OPEN "I",2,"FILE2"
100 OPEN "I",3,"FILE3"
110 INPUT #1,A
120 INPUT #2,B
130 INPUT #3,C
140 PRINT A,B,C
150 CLOSE
OK
RUN
 1          3          3
OK
FILES
  FILE1   FILE2   FILE3
OK
```

Better than Dirt Bikes?

Just a quick note with a few attaboys and one or two complaints. First, nice magazine; keep it up. Good balance on the hard vs soft. Kilobaud Klassroom is great for us new guys playing catch-up.

DRC of Garland gets the teleportation award. Good price on the 2708 in September issue, right? Phone in, give credit-card number and have the chips in 48 hours! That's the way to keep customers!

One attaboy to Omaha Computer store, for trying really hard. Just so-so in the speed department, but really thorough, and sticks by the little guys.

One pataback for PolyMorphics Systems for a super system for the beginner. Enough elementary stuff to keep the interest up between op-code sessions, but looks flexible enough to expand with a pro. Very poor documentation on assembler. Not usable without at least a year of 8080 experience or help.

Been running since July and have IC tracks in both arms. What a gas. Better than dirt bikes! Well BASIC balance saver is calling.

Keep it up.

Mick Topping
Omaha NE

A CPA for Consultations

Edward Campbell's article, "Starting a Business? . . . pit falls to avoid," in *Kilobaud* No. 9 was a fairly good primer for potential entrepreneurs. Although Mr. Campbell talked about forms of organization — proprietorship, partnership and corporation — he failed to mention that there are federal and state income-tax implications involved. Consulting a certified public accountant would probably be a better course of action then obtaining a business consultant. The CPA not only would do what the

business consultant would do, but he would be most helpful in determining the tax aspects of the business.

W. Stephen Rice
Dunbar WV

Photo Data Needed

As a professional photographer and amateur programmer, I need some help. I have written a program that should translate zone-system parametrics from one person's camera system to another's. However, due to a lack of data base, I have not been able to give this program a thorough test. It seems to work, but I need more zone-system data to be sure.

So, if you are interested in the possibility of getting some more film-developer combination data, and at the same time helping out a puzzled programmer, please send me the following information:

- A chart of your developing times and resulting ten zone densities.
- The film speed (ASA) of the combinations you send.
- The developing temperature.
- The film format.
- The film type and brand.
- The developer and any special information on its use, such as dilution, whether it is diluted with water or a chemical solution, or if any chemicals are added to the developer.

If you will include a stamped, self-addressed envelope, I will try to send you at least one new set of parametrics, and will (sooner or later) send you the BASIC program listing and some notes about what has happened with it so far.

Michael P. Avery
725 Foxboro Lane
Dallas TX 75241

Computer Stores

We've read with interest Ken Barbier's article in the November 1977 issue. The Computer Store maintains over $100,000 in inventory and a depth of computer systems, ready to run with applications software sufficient to meet the needs of any customer for any system.

The customer's access to fresh coffee is unlimited (they've gotta bring their own donuts, though).

We invite Mr. Barbier or any of your thousands of readers to visit our Burlington MA store and chat with any of our helpful customer-service people, try any of the five different computers we feature (Data General microNOVAs, Wang, Cromemco, Compucolor and Apple II), or resist the temptation to buy computers, peripherals, supplies or books we stock and sell.

Paul C. Conover
The Computer Store, Inc.
Burlington MA

Speed Tips

Two things in the October issue prompt me to write: the book reviews and the article on timings.

Mr. Enderle told us about *Programming Proverbs*, and I feel his assessment is accurate. Many beginners need models on which to form good habits. I would suggest two similar books: *Programming Style, Design, Efficiency, Debugging, and Testing,* by Denny Van Tassell, and/or *The Elements of Programming Style*, by Brian Kernigham.

My comments about the timing comparisons are based on my own experiments on just one machine. First, to enable me to start timing conveniently, for example, when the second-hand is on 12, I have two lines like 100 PRINT "READY"; and 150 INPUT X just before the section to be timed. When the program is run, I can enter any value for X and hit RETURN just as the second hand hits 12. This seems easier than noticing when the final T in START is printed.

Second, it is instructive to try various forms of the same statement to see how your particular system does in terms of speed. To illustrate to one of my classes the complex and slow algorithm used for exponentiation, I put LET K = 2 ↑ 3 in a 1 to 250 loop. This took 13 seconds. Changing the computation to its equivalent 2*2*2 reduced the time to about 2.1 seconds. For exponents above 3, the savings would, of course, be less. Anyway, this might get your readers thinking about other timing checks, and "prove" some of the advice in the aforementioned books on good programming.

Jim Gross
Sheboygan WI

100-Pin Board, S-100 Compatible?

I have just finished reading issue No. 10. In my opinion, it is your best effort yet. I especially liked the article titles "Put Your Imsai on the Rack" and "Beware the Altair Bus."

I have an Imsai 8080 and have plans for several non-IMSAI, S-100-bus-compatible PC boards. Obviously, the article on S-100-bus compatibility was timely for me. It has raised a few questions, however, and I hope you will be able to offer some solutions that may benefit all S-100-bus users.

How is it possible that a manufacturer can advertise a PC board to be S-100-bus compatible when, in reality, it may not be? Should your publication require a little more detail for an advertisement then simply S-100-bus compatible? Why not require the manufacturer to include the CPU board and memory board, or boards, their product has been tested with, for determining its compatibility?

I feel we should leave the responsibility for a PC board's compatibility with the design engineer, not the hobbyist or businessman using the product. Should we be expected to modify a standard design system, costing perhaps $1000 or more, so we can use a so-called standard PC board? If so, something is not quite right. I would rather think I could buy any S-100-bus-compatible product, perform any special hookup per the manufacturer's ad, power up and run. Bus modifications should be acceptable as long as we know about them before we purchase the product. I'm an optimist, right? There would be no unexpected bus modifications, and I would be a happy consumer once again. That sounds better to me then becoming the victim of a product.

Come on, *Kilobaud*! Give us little guys a hand with this problem. You have all the right connections and the clout to go with them. All I have is typewriter and paper.

You are doing a super job with *Kilobaud*. I trust you will keep it up.

Allen L. Cox
Prattville AL

Program Submission Points

I have a request to make. Please tell us in your programs how much memory and what language was used.

A column on changing from one BASIC to another (such as "change" used in some BASICs) would help not only me, but others who can't afford or don't have access to all the BASICs.

John W. Neel
Apopka, FL

For Fun and Profit

Glad to see you helping out the businessman who is interested in the non-technical side of computers. I am interested in economically advantageous uses of computers as well as the technical details and leisure-time purposes such as "programming for the fun of it." The economic facet of computers is certainly an important one. *Kilobaud* will continue to be read.

Bob Cave
Irving TX

PUBLISHER'S REMARKS

(from page 6)

yet. If you have worked out some of your own programs, would you be willing to share them with other *Kilobaud* readers? If you've bought programs, perhaps there is a way these can be made available. If you've located some "public domain" programs which are of use, give us a hint on those too.

Video Disk Demise

There was some discussion in other magazines about the possibilities of using the video disk systems for software distribution. It now appears that the challenge of the videotape recorder (VTR) has been too much for the disk developers. RCA seems to have indefinitely put off their work in this area . . . as has Zenith, Telefunken and Decca. Now we hear that Phillips has put off for at least a year their plans to introduce a disk. With VTR prices dropping, it seems unlikely that disks will make it.

A two-hour videotape cartridge can store a lot of computer data. As VTR systems get more popular we may see them being pressed into this service. Most programs can be stored quite easily on audio cassettes, so the need for the massive storage capacity of VTR will probably be more as a replacement for the several-thousand-dollar tape drives than for program transfer. The VTR might be a good device for storage of large amounts of data. How about it?

The VTR systems are quite good, by the way. The $1000, plus or minus, price may slow you down a bit, but once you have one you are free of the restrictions put on you by the TV network planners. You can watch one TV show and record another . . . or set your recorder to tape the programs you would otherwise miss while away. The flexibility of the systems will be developed . . . probably with computer programming of the channels and times you want to record. But the picture quality is already quite good and the color acceptable.

Are We Looking for You?

We are looking for people who demand a lot out of life . . . who are not satisfied with an ordinary job, watching television, or perhaps playing around with a small computer system for fun. We are looking for people who enjoy learning, and have a commitment to themselves to succeed. They can surmount frustration and an occasional wrong decision. They believe in themselves and will fight for what they want, getting things done regardless of obstacles. They will do this without compromising their honor and ideals.

A high-pressure business will bring out the best in these people, and present challenge, competition and excitement. Such a business offers opportunities for substantial income, yet since it is starting small there is almost unlimited growth potential for both people and business.

Kilobaud, with its expanding magazine, computer books, program sales and other plans, is an ideal place for success-oriented people to grow. In fact, considering the potential of the small-computer industry, seldom has there been an opportunity such as this to be in at the very beginning.

We have openings for writers, editors, technicians and BASIC programmers. There are openings for sales and marketing people, electronics draftsmen, and for all of the other elements that go into producing magazines, books, programs, etc.

We're located in the mountains of southern New Hampshire, a state without income tax or sales tax (so far). Living is more country-style, even though the benefits of a big city (Boston) are only a little over an hour away. Boston has a major airport, shows, restaurants and shopping. New Hampshire has outdoors – for hiking, skiing, beauty and clean air.

The work here is unstructured, leaning more toward personal responsibility than heavy management. It's relaxed and interactive.

If all this sounds good to you, write, giving your background and some reasons why you think you'd like to be with this group . . . and what you would like to accomplish. If you are thinking about a microcomputer-technician position, make sure you are quite familiar with a half-dozen or so microcomputer systems – using them, servicing them, using various I/Os with them, etc. If you are a BASIC programmer be sure you have experience with several versions of BASIC and know what you are doing. Once you get involved here it will be a little late to shore up your fundamentals.

The only serious restriction is on the use of tobacco and other drugs. We enjoy clean, pure air, and there is no smoking permitted here. If you have a tobacco addiction we sympathize with you and wish you well . . . somewhere else.

Peterborough is a town of about 4000 people. It is the business center for the region, with two shopping plazas and a downtown area, too. It's a rapidly growing center with two large mail-order houses – Brookstone for tools and Eastern Mountain Sports for sporting goods – plus several other major businesses taking advantage of the low taxes and wonderful climate.

Will we be able to keep up with the growth of the microcomputer industry? Will we be able to take program sales from zero to hundreds of millions of dollars a year? It's possible . . . and we need people to help make this happen . . . and to enjoy it all. Write.

AROUND THE INDUSTRY

(from page 8)

the System 8813. Bob is shown in Photo 4 with his new pride and joy. He has a lot to be proud of, too. He has come up with a sophisticated, yet easy-to-use, operating system. His primary goal was a system that would be as easy to use for the novice as the professional. This is evident in a lot of the little things he thought of while designing the system. For example, how many systems have you seen that dictate a minimum of five or six characters for a file name? Sometimes it can be downright hard to come up with a name working under those constraints . . . and, of course, when you do it's an abbreviation, mnemonic or acronym. Why not let the file name be 31 characters long? That's the way Bob felt about it.

If the user wants to run a program (any program), he simply enters the file name of the program . . . and it runs. That may sound terribly simplistic, but it's not that simple with *all* operating systems. If it's a BASIC program, then BASIC is automatically loaded before the program. The user can specify any program as an "initial file" so that it will be automatically loaded, and run upon power-up. Needless to say, any operating system written to support a 3-disk system is extensive and will require more space than I have here to describe it adequately. I've pointed out just some of the features that reflect Bob's user-oriented approach . . . that concept is perhaps the most important.

The BASIC used in the system is a modified version of Poly's AOO that includes

their PLOT statement for generation of graphics. A rather significant applications program, an inventory control package, should be ready for release by the time you read this. Prices start at $3250 for a single-drive system with 16K of memory, keyboard and monitor, and go up to $4430 for a three-drive configuration (which requires an additional 16K, for $495, for a total of 32K). The cost isn't low by any means, but if I were putting together small business systems and wanted something I could take out of a box, plug in and run (without having to hassle with hardware and software interfacing), I'd certainly give it serious consideration. (PolyMorphic Systems, 460 Ward Dr., Santa Barbara CA 93111.)

Hal Singer's Young Men

Last month, I mentioned Hal Singer (math teacher) and David Bryant (student) at Cabrillo High School in Lompoc CA (where we hold our monthly meetings of the Micro-8 Computer Club). Another of Hal's exceptional students is Jerry Nix (Photo 5, seated on the left, with his eyes closed). At a recent Micro-8 meeting, Jerry demonstrated a rather remarkable word-processing system (which you will be able to

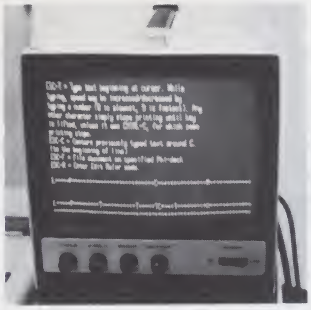


Photo 6.

buy through the Digital Group in the near future). The system works in conjunction with Digital Group's dual-Phi-deck system and its operating system, PHIMON, written by David (facing the camera in the right rear of Photo 5).

Photo 6 is a sample of what Jerry's software can do. With the Edit Ruler shown across the bottom of the screen you can specify the left margin (L), the paragraph identation (P), up to twelve tabs (T), the center of the page (C) and the right margin (R). I hope we'll get a review on this package in an upcoming issue.

Dealer's Association

Portia Isaacson, vice-president of Binary Systems Corporation, has been nominated to be chairperson of a committee to form the first national association of independent computer store owners. The initial exploratory meeting of some 40 store owners was held in Dallas last June at the National Computer Conference convention.

An organizational plan for the association, which will be known as the "Computer Retailer's Association," was presented by Los Angeles attorney Kenneth Widelitz. For further information, contact Widelitz at: 10960 Wilshire Boulevard #1504, Los Angeles CA 90024.

Photo 5.

THE HEATHKIT FORUM

(from page 15)

given that Heath has built a microprocessor trainer around it.

Ren Colantoni joined John and Raymond in criticizing the use of octal rather than hexadecimal notation. I like octal for machine-language programming of the 8080 since most of the instructions fall naturally into three-bit fields indicating which of eight possible variations are involved. These include: restarts; conditional jumps, calls and returns; and instructions involving the registers.

It is true that hexadecimal notation saves a third of the keystrokes required for program entry. So you may want to write a program to convert the H8 front panel from octal to hex. Heath has made this easy by including in the monitor ROM a subroutine to set the A register to a hexadecimal value when a key is pressed. The value corresponds to each of the numbered keys, with A-F produced by the six unnumbered keys that follow 9 in clockwise order.

My H8 was delivered the day after Labor Day, but I haven't been able to do anything with it as I bought it without memory. I don't like the idea of paying Heath for 4K of memory what other companies are charging for 8K. Fortunately, I've heard reports that other firms are planning memory cards for the H8. Meanwhile, I've ordered enough memory chips to make my own 256-byte memory. I'll let you know how it turns out. And you let me know what you've been up to. Write:

The Heathkit Forum
c/o Charles Floto
267 Willow Street
New Haven CT 06511

Heathkit is a registered trademark of the Heath Company, Benton Harbor MI.

LEGAL/BUSINESS FORUM

(from page 16)

playing the package of any product on which the text of the written warranty appears.

The effect of the Presale Availability Rule will be dramatic. The rule will allow you to compare warranties more readily than has been possible in the past. For that reason, the warranty should become a much more important factor in your purchasing decision. That should provide incentive for the manufacturer to give greater consideration to his warranty.

It is the duty of the warrantor to provide retailers with the warranty materials. It is the obligation of the retailer to make sure the materials are properly displayed and kept up to date.

Perhaps in the long run such costs will be passed on to the consumer. I submit that the benefit to the consumer in being able to shop for warranties more than makes up for the added printing costs.

As yet, the FTC has not

(continued on page 112)

Robert C. Brehm
Promedics Data Corporation
1032 Elwell Ct., Suite 240
Palo Alto CA 94303

Business Applications

a system for the professional

Last month, I mentioned small-business applications in my editorial. This month, we re starting a new section in KB aimed toward the small businessman. Our objective is to keep him informed of the latest developments in small-business systems (hardware and software) and let him know what is available. Bob's article is the first in this series, and it does a fine job of meeting that objective for the medical profession.

Quite frankly, Bob's article could be construed as a multi-page advertisement for Promedics Data Corporation. If that's the case then Bob and his company are doing OK, especially since I also paid him for the article.

Two facts remain: 1. I asked Bob to write the article and he came through with a fine job; 2. The article informs the medical profession (and other professionals) about the Promedics system. And, as I said earlier, that is our objective . . . because they don't have any other publication from which to get such information.

Remember, I'm also looking for user-written articles. — John.

Today's physician, dentist, lawyer or small businessman has the opportunity to improve customer service, increase cash flow and decrease operating costs by obtaining accurate and timely financial information from a turnkey microcomputer system. What are these systems, what can they do and how much do they cost?

This article will answer these questions by describing a turnkey system now being sold to dentists and physicians by Promedics Data Corporation. The article describes system requirements, available hardware and how it is used, software necessary to perform the business functions of a small to medium-size office, and the training and maintenance required for computer systems support. All of these areas are important considerations when deciding what system you need, where to purchase it and how to use it.

Background

Today, a physician or dentist must not only be skillful at his profession, but he must also be skillful as a businessman. Several trends within the business portion of a practice have made it difficult to effectively perform this duality of roles. These trends include: increased insurance processing, larger practices, multistaffed offices, and increased paperwork and external reporting.

The Promedics Patient Accounting System can help the professional keep pace with these accounting requirements by greatly increasing the speed, accuracy and ease with which patient service information can be processed.

The small turnkey microcomputer system described here offers several benefits over a larger minicomputer or full-scale computers. One prime factor is cost. A microcomputer system with software generally sells for ten to fifteen thousand as compared to a mini at $20,000 to $40,000 or a large computer at $100,000 and up. Microcomputers are easier to use than larger computers since cumbersome job-control language and procedures are avoided. Microcomputers tend to be more compact and take up less space than their larger counterparts. They are also less expensive to run, easier to maintain and quieter to operate.

The Patient Accounting System (PAS) implemented on the PAC 1 microcomputer system has several key

features that are useful in a professional office. The software will generate patient statements and insurance forms, record and report services rendered, accounts receivable with aging, service analysis and patient statistics. It also can report any patient information, such as current balance, at the stroke of a key. The software is easily learned and provides a standard for all personnel to use. It is also highly flexible, since it is written in a high-level language, and can be easily changed to meet individual-practice requirements.

With this background in mind, let's take a closer look at a typical practice and examine the system requirements in more detail; then we can review a general solution using a microcomputer system.

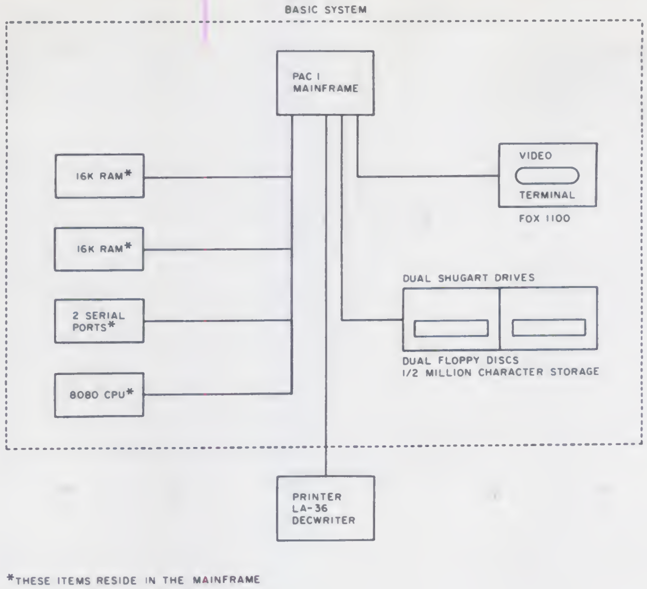


Fig. 1. System hardware configuration.

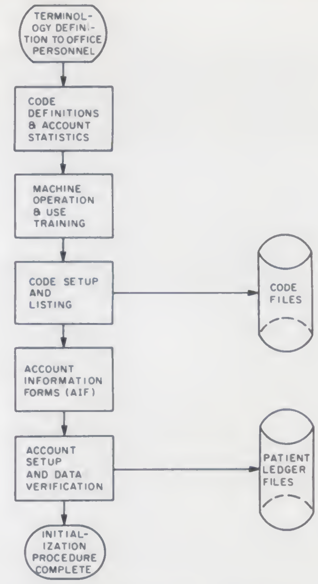


Fig. 2. Data-base file initialization procedure.

General Need and System Requirements

A typical medical or dental practice often consists of two or more doctors, a hygienist or registered nurse, an office manager and several dental or medical assistants. An average patient base for an office can easily reach 2000 people. Keeping track of all services provided and by whom is usually a difficult, time-consuming task. With today's ever-increasing insurance benefits, it is not unusual for a practice to have 60-75 percent insurance billing, all of which requires an insurance form or two. Moreover, since our society is accustomed to paying for services on an installment basis, it becomes necessary for the medical or dental office to have an accurate and timely report of all accounts receivable, with an aging of each account.

The accounting procedures and practice management techniques utilized in professional offices have not followed the advances in medical and dental technology over the last 30 years. Many practitioners are using the newest techniques and materials for filling a tooth, yet their office managers are still using the "pegboard" accounting system developed before many of them were born! The information requirements by outside agencies like insurance companies has increased to the extent that many practices often add an extra person just to process insurance forms and patient statements.

Many of these problems and information requirements can easily be handled by an inexpensive turnkey microcomputer system. Based on the above facts, the requirements of such a system can be summarized as follows. The system must:

- Be easy to use, require little if any maintenance, take up a small amount of space and be quiet running.
- Communicate and direct the user in a language familiar to the operator (e.g., in English phrases using medical or dental terminology).
- Be cost effective, and in general cost less than hiring an extra person.
- Use accounting techniques parallel to those currently used in order to reduce errors and confusion when converting patient data to computer files.
- Store all insurance and ledger card information in a concise manner and allow any patient's data to be accessed quickly.
- Provide daily reports of new accounts, charges and receipts and doctor services performed.
- Be able to accommodate multiple providers of service to a patient base that is constantly changing and growing.
- Automatically generate patient statements and insurance forms at any time of the month or billing cycle.

CRT or Video Terminal: Cathode Ray Tube, a video screen much like a TV set. This device is used in your computer system to enter, change and delete data stored in the computer's memory banks.
Floppy Disk: An 8-inch-square piece of plastic-like material that has a magnetic coating similar to that used on a cassette tape. Its purpose is to store large amounts of data in a small removable package. It is organized into 77 tracks similar to a phonograph record but without the grooves. Each floppy disk can store approximately 250,000 characters of information or the equivalent of 500 pages of information. Floppy disks are used in a floppy disk drive. The PAC 1 system utilizes two drives.
RAM or Memory: Random Access Memory, used to store programs and data. RAM is measured in terms of number of characters (or bytes) of storage capacity. The PAC 1 system uses 32,000 memory-storage bytes.
Programs: Specific instructions in computer language that tell the machine what to do. For example, a set of instructions can be written to make the computer print address labels using data stored on a floppy disk.
CPU: The central processing unit is the heart of the computer and is the electronic part that performs all the computer logic that makes your computer work.
Printer: Used for hard copy of reports or for printing statements or mailing labels, etc. It can be a high-speed or low-speed type, depending upon your requirements.
Software: The set of programs which control the computer and make the different pieces of hardware interact with each other.
Hardware: All the mechanical parts of the computer system, including the mainframe or CPU, the printer, the floppy-disk drive and the video terminal.

Table 1. Definition of terms.

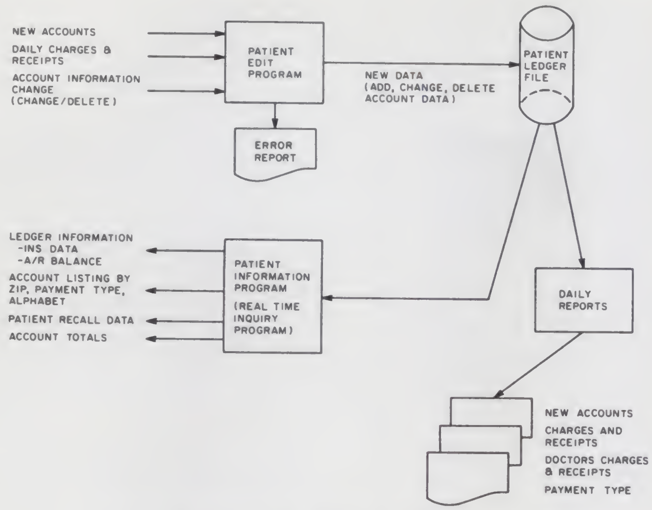


Fig. 3. Software flow diagram of daily routines.

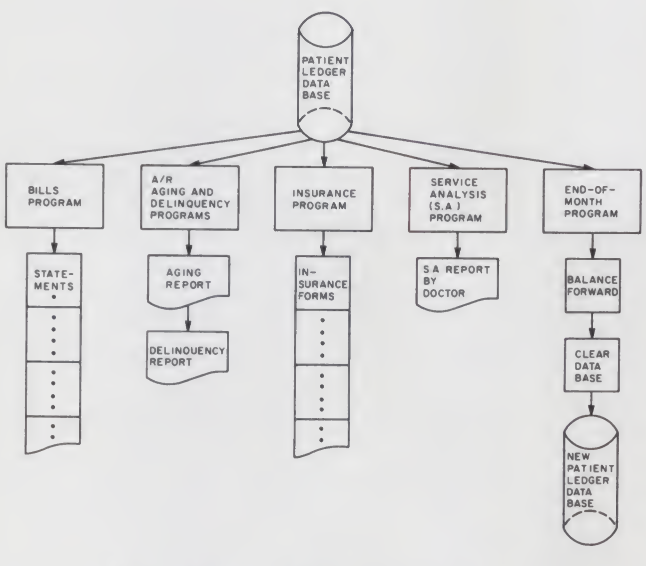


Fig. 4. Information flow for monthly reports.

- Provide accounts-receivable reports, including an aging analysis, along with other important management reports.

With these requirements in mind, let's look at several aspects of a microcomputer system currently being installed in medical and dental offices on the West Coast.

However, before we delve into a semitechnical discussion, it is helpful to define a few of the terms used in the rest of the article. These are given in Table 1.

Hardware

A schematic of the basic building blocks of the PAC 1 hardware configuration is shown in Fig. 1.

The computer mainframe utilizes the economical, reliable and powerful 8080 microprocessor, together with 32K of random access memory (RAM); two serial ports for transferring information to and from the CPU, video display and hard-copy printer; and a floppy disk interface card. This hardware features solid-state reliability and high-speed operation, which allows it to perform thousands of calculations per second. Due to the modular design of the plug-in computer cards, the system is easily expandable in terms of more memory, faster terminals or larger mass storage capacity.

The dual floppy disk used in the basic system incorporates the well-known Shugart drives used throughout the world. Each floppy disk can store a quarter-million characters using the industry-standard IBM 3740 format. From a practical standpoint, this means that all ledger and insurance information for 2000 patients can be stored on-line at one time. Within several months, dual-density technology should allow the disks to store approximately 3500 patients on-line. Up to four drives may be used on this system for a storage capacity of 7500 patients on-line. These floppy disks were chosen for their fast access to patient information, their mass storage capability and their hardware reliability.

The video display terminal used in the PAC 1 system incorporates a microprocessor inside, thus making the system more reliable due to fewer parts. The terminal's 12-inch screen displays 24 lines of 80 characters, which may be transferred from the CPU mainframe up to speeds of 9600 baud. The video display terminal is used for data input and patient-information recall. It normally runs at 2400 baud, but may be adjusted to any convenient operator speed.

The LA-36 printer produces hard-copy reports (charges and receipts, new accounts, accounts receivable) and prints patient statements and *insurance* forms. The printer operates at 30 characters per second and can imprint up to six copies. This printer is cost effective, reliable and easily fulfills the output requirements of a small to medium-size dental or medical office.

All of the equipment in the system is color-coordinated in eggshell white to match the decor of most professional offices. In addition, the equipment is burned in and tested in all aspects

SERVICE DETAIL	SERVICE SUMMARY	DEFAULT $	DESCRIPTION
1	1	.00	BALANCE FORWARD
2	17	.00	ERROR -- OVERCHARGE
3	17	.00	ERROR -- UNDERCHARGE
4	17	.00	ERROR -- OVER PAYMENT
5	17	.00	ADJUST CHARGES
6	17	.00	ADJUST CREDITS
7	17	.00	DEBIT -- WRITE OFF
8	17	.00	CREDIT -- WRITE OFF
9	17	.00	INSURANCE CREDIT
10	17	.00	OTHER ADJUSTMENT
11	19	.00	CASH PAYMENT
12	19	.00	CHECK PAYMENT
13	19	.00	C.D.S. PAYMENT
14	19	.00	MEDI-CAL PAYMENT
15	19	.00	WELFARE PAYMENT
16	19	.00	MEDICAID PAYMENT
17	19	.00	MISC INSURANCE PAYMENT
18	19	.00	OTHER PAYMENT
20	2	8.00	OFFICE VISIT
30	2	15.00	PROFESSIONAL VISITS
40	2	15.00	SPECIAL CONSULTATION
49	2	12.00	PROPHYLAXIS-CHILD
50	2	15.00	PROPHYLAXIS-ADULT
62	2	15.00	FLUORIDE TREATMENT
80	2	7.00	EMERGENCY TREATMENT
110	2	4.00	SINGLE FILM
111	2	1.00	ADDITIONAL FILM

before it is delivered to help insure trouble-free operation from the date of installation. Furthermore, the hardware configuration and all its individual components are designed to work in conjunction with the software. Consequently, when a problem does occur, it can easily be pinpointed to either the software or hardware since a standard hardware configuration is used.

The same hardware system is also available for business systems or for the entrepreneur who desires to do his own programming using an ALGOL-like high-level language compiler/interpreter available from Promedics. The hardware system easily satisfies the requirements of a one- to ten-person medical or dental practice and most small businesses with 10-100 employees.

Software

The software supplied with the PAC 1 for medical and dental applications implements an entire Patient Accounting System (PAS) on the microcomputer. A software flow diagram of the entire system is shown in Figs. 3 and 4. The system is written using a high-level language for ease of programming and support purposes.

When the PAS is being implemented, several training and implementation phases are followed until a routine procedure is established.

These initial phases of training, given to key office personnel, cover definitions and computer terminology, which may be unfamiliar to first-time computer users. Once terms are learned and the general characteristics of the system explained, a careful study is made of the current patient-accounting system to determine number of active accounts, average patients per account, service codes, doctor codes, diagnosis codes, average transactions per account and additional information required for setting up data file sizes and

112	2	22.00	DENTURE SERIES FILMS
113	2	10.00	INTRA-ORAL FILMS
114	2	15.00	LATERAL JAW-ONE FILM
115	2	20.00	LATERAL JAW-TWO FILMS
116	2	7.00	BITE WINGS & EXAM
117	2	9.00	BITE WINGS-4 FILMS
118	2	1.00	BITE WINGS, ADDITIONAL FILM
125	2	25.00	PANOREX FILM
126	2	25.00	CEPHALOMETRIC FILM
150	2	25.00	BIOPSY OF ORAL TISSUE
160	2	15.00	MICROSCOPIC EXAM
200	3	12.00	SINGLE EXTRACTION
201	3	10.00	ADDITIONAL EXTRACTION
202	3	20.00	SURGICAL REMOVAL OF TOOTH
220	3	5.00	POST-OPERATIVE VISIT
230	3	25.00	REMOVAL OF TOOTH
231	3	45.00	REMOVAL OF TOOTH (2)
232	3	55.00	REMOVAL OF TOOTH (3)
250	3	35.00	ALVEOLECTOMY
252	3	25.00	ALVEOLECTOMY (2)
256	3	42.00	ALVEOPLASTY
257	3	35.00	PALATAL TORUS REMOVAL
258	3	35.00	MANDIBULAR TORI REMOVAL
259	3	40.00	EXCISION
278	3	20.00	MAXILLIARY SINUSOTOMY
280	3	35.00	EXCISION OF CYST
281	3	50.00	EXCISION OF LARGE CYST

SERVICE SUMMARY	DESCRIPTION
1	MISC SERVICES
2	DIAGNOSTIC
3	ORAL SURGERY
4	PERIDONTICS
5	ENDODONTICS
6	RESTORATIVE
7	CROWN & BRIDGE
8	PROSTHETICS
9	ORTHODONTICS
10	BROKEN APPOINTMENTS
12	DENTAL SUPPLIES
15	BALANCE FORWARD
17	ADJUSTMENTS
19	PAYMENTS
20	COMPUTER SERVICES

DIAGNOSTIC CODE	DESCRIPTION
1	FIRST TIME VISIT
2	DENTAL PROCEDURE
3	ORTHODONTIC PROCEDURE
4	OUT PATIENT SERVICES
3	MISC SERVICES
5	COMPUTER SERVICES

PAYMENT TYPE	FORM #	DESCRIPTION
1	0	CASH PAYMENTS
2	1	MISC INSURANCE
3	1	BLUE CROSS/SHIELD
4	2	MEDICAID
5	3	MEDI-CAL
6	3	C.D.S.
7	0	WELFARE
8	1	UNICLAIM
9	1	OTHER
15	0	CREDIT CARD PAYMENTS
20	0	MISC BILLING

DOCTOR NUM	DOCTOR NAME	INITIALS	DEGREE	LICENSE NUMBER	PHONE NUM
1	JOHN B. ANDERSON	JA	DDS	D123723	555-1716
2	RONALD R. EDWARDS	RE	DDS	A18772	555-1213
3	KATHY JACOBS	KJ	DH	34223	555-1213
4	BARBARA SNYDER	BS	RDH	55222	555-1213
5	SANDY WALSH	SW	RDH	99276	555-1213

Fig. 5. Sample codes set up for a dental-office application. Each code number and definition is defined by the user for his own requirements. This means that different service codes, payment codes, etc., may be set up for physicians, dentists, optometrists, etc. These codes are used in the various reports and also for input responses.

Keyboard used for data input and recall of patient data on video screen.

Hard-copy printer used to print patient statements, insurance forms, management reports and mailing labels. Auxiliary keyboard provides backup to video terminal keyboard.

parameters. After these factors are defined, the data base is set up on the computer by use of actual data, which is then verified for accuracy before further proceedings are started.

Fig. 2 illustrates the initialization procedure, in which doctor/hygienist information, payment codes, service codes, diagnosis codes, special message codes, expense codes and recall codes are defined and put into the data base resident on the floppy disks. This procedure is usually done once at the beginning of operation of the system, but any code can be changed at any time. Quite often, service codes are set up according to AMA or ADA standards or those employed by insurance companies. Fig. 5 shows a typical listing of various codes used in a dental office.

After all codes are defined and entered, an Account Information Form (AIF) is filled out for each account (usually the head of the household), with factors such as name, address, phone number, sex, birth date, insurance-policy data, patients in family and any current balance in effect when the account is set up. The AIFs bring together information concerning the account, which is often stored in many places. The process of data accumulation often eliminates several filing cabinets of excess paper and also sets up a standardization for data collection probably nonexistent before. The account information is then entered into the computer, the initialization process is completed, and the computer is now ready to begin routine daily processing.

A flow diagram of daily processing routines is shown in Fig. 3. During a normal day, the receptionist or assistant would be adding, changing or deleting account information, including setting up new accounts, charging services, recording collections and making account adjustments. At the end of the day, an error report is generated showing invalid codes or services entered — these may be changed and reentered. At any time during the day, all patient information is available for instant recall through the video terminal. This includes such information as account balances, insurance type, totals for user-selected attributes and recall dates, to name a few. In addition, the traditional daily charges-and-receipts report is also generated along with reports showing all new accounts, payment-type analysis and charges and receipts broken down according to doctor. All of these reports are useful for standardizing patient financial history, for documenting audit trails to keep your accountant happy and for developing a sense of how well the practice is being run and what improvements or changes may be necessary. Many of these reports are optional and may be run on an "as needed" basis. Sample daily reports are shown in Figs. 6 through 11.

At the end of the day, the data disks are copied over to the next day's data disks for security purposes. In this manner, should something fail during a given day, the maximum exposure to data loss is one day, since the user has a copy of the prior day's data on another disk. Usually two sets of disks are used daily and copied over to the next day's disks for use on the following day. Thus, five disk sets are needed for Monday-through-Friday operation.

NEW ACCOUNTS

JUL 22, 1977

NO	ACCOUNT	NAME	ADDRESS	TELEPHONE	ACCT CTRL	TYPE
1	10009	FERNWOOD, FREDDIE	1095 FAIR OAKS DRIVE MENLO PARK, CA 94025	415 854-2319	0	1 6
7	10031	HOLMS, O.W.	66 DOWNING STREET MILPITAS, CA 94232	415 545-1919	0	1
10	10032	AHL, NEWETT	2323 ROCHESTOR AVE MENLO PARK, CA 94025	415 553-1998	0	1

Fig. 6. This report, run on a daily basis, lists all new accounts set up during the day. It provides a written verification of new accounts and their pertinent data.

CHARGES & RECEIPTS REPORT

JUL 22, 1977

NO	DATE	ACCT#	FRP NAME	CASE#	PAT	TYPE	DR	DIAG	TRAN	CHARGE	RECEIPT	PRIOR BAL
3	07/23/77	10009	FERNWOOD, FREDDIE	1	FF	1	1	2	20	8.00	0.00	0.00
4	07/23/77	10009	FERNWOOD, FREDDIE	1	FF	1	1	2	62	15.00	0.00	0.00
5	07/23/77	10009	FERNWOOD, FREDDIE	1	FF	1	1	2	125	25.00	0.00	0.00
6	07/23/77	10009	FERNWOOD, FREDDIE	1	FF	1	1	2	12	0.00	25.00	0.00
9	07/23/77	10031	HOLMS, O.W.	1	OW	1	3	2	1	69.00	0.00	0.00
12	07/23/77	10032	AHL, NEWETT	1	NA	1	4	1	1	12.50	0.00	0.00
TOTALS										129.50	25.00	

Fig. 7. The Charges and Receipts report is a daily report listing all services performed and all receipts or adjustments received from either the patient or an insurance company.

METHOD OF PAYMENT REPORT

JUL 22, 1977

TYPE	DAY		MONTH		YEAR	
	CHARGES	RECEIPTS/NET ADJ	CHARGES	RECEIPTS/NET ADJ	CHARGES	RECEIPTS/NET ADJ
1	210.50	1113.67	4046.50	538.67	4046.50	538.67
2	85.00	107.00	568.00	7.00	568.00	7.00
3	0.00	0.00	629.00	0.00	629.00	0.00
4	59.00	339.00	1239.00	339.00	1239.00	339.00
5	0.00	550.00	1057.00	250.00	1057.00	250.00
6	0.00	455.00	1785.00	155.00	1785.00	155.00
7	132.00	0.00	1160.00	0.00	1160.00	0.00
8	0.00	210.00	912.00	210.00	912.00	210.00
9	0.00	0.00	28.00	0.00	28.00	0.00
TOTALS	486.50	2774.67	11424.50	1499.67	11424.50	1499.67

TYPE	DESCRIPTION
1	CASH PAYMENTS
2	MISC INSURANCE
3	BLUE CROSS/SHIELD
4	MEDICAID
5	MEDI-CAL
6	C.D.S.
7	WELFARE
8	UNICLAIM
9	OTHER
15	CREDIT CARD PAYMENTS
20	MISC BILLING

Fig. 8. This report, which can be run on a daily basis if necessary, breaks down charges and receipts to specific payment types in order to show you which type is paying the least, the most, etc.

DOCTORS' CHARGES, RECEIPTS/ADJ REPORT

JUL 22, 1977

DOCTOR	DAY		MONTH		YEAR	
	CHARGES	RECEIPTS/NET ADJ	CHARGES	RECEIPTS/NET ADJ	CHARGES	RECEIPTS/NET ADJ
1	66.00	916.67	2684.00	916.67	2684.00	916.67
2	132.00	1021.00	3778.00	421.00	3778.00	421.00
3	132.00	482.00	1439.00	207.00	1439.00	207.00
4	71.50	355.00	2953.50	155.00	2953.50	155.00
5	85.00	0.00	570.00	-200.00	570.00	-200.00
TOTALS	486.50	2774.67	11424.50	1499.67	11424.50	1499.67

CODE	DOCTOR/HYGIENIST
1	JOHN B. ANDERSON
2	RONALD R. EDWARDS
3	KATHY JACOBS
4	BARBARA SNYDER
5	SANDY WALSH

Fig. 9. This report, available on a daily basis, breaks down charges and receipts according to producer to illustrate the relative differences between producers.

This procedure may cost a little more, but it is inexpensive when compared to reconstructing an entire data base, should the power fail and wipe out a data disk.

All of the above procedures are followed on a daily basis until the end of the month, when additional program functions are run as discussed next.

During end-of-month processing, diagrammed in Fig. 4, several additional tasks are performed and reports are generated. First, the accounts are usually sorted by zip code for ease of mailing, and then the patient statements are printed in the form of speedy mailers, wherein all information is printed on a six-part form that includes a mailing envelope, patient statement and return envelope. This simple design allows all billing data to be passed through the computer only once and eliminates end-of-the-month envelope-stuffing of patient bills. In this case the mailers are simply run through a postage machine and then dropped into the mail. The first copy of the six-part form is kept in the office as a reminder of all patient statements sent.

After statements are processed at month's end (they may also be generated anytime during the month), the insurance forms are printed using standard insurance form formats for Blue Cross, Medicare, Medicaid, California Dental Service, etc. This generation of forms takes the paperwork burden from medical or dental assistants and places it on the computer, thus allowing the assistants to work on medical problems for which they were trained.

When bills and insurance forms are done, several management-type reports can be run. These include an accounts-receivable aging report sorted by payment type (e.g., cash, credit card, insurance type 1, insurance type 2, etc.) and either alphabetical order or account-number order. A typical accounts-receivable report for a payment type 1 is shown in Fig. 12. This report gives each account, when last paid, how much and the age of balances.

Another report, generated as part of the accounts-receivable data is the delinquency report, which shows all accounts whose balances are over either 90 days or 120 days (selectable). The report is shown in Fig. 13. This is an "action" report used to contact the overdue account and request payment.

Perhaps the most important report generated at month's end is the service-analysis report shown in Fig. 14. This report categorizes services by service summary code and presents both number and percent and dollar and percent generated by each service, each category and also total. This report is normally broken down according to doctor, with a grand total given for all doctors. By careful study of this report, a doctor can determine which services he might delegate to auxiliary

```
ACCOUNT:  10023      MAXWELL, ANTHONY                                     7/23/77

SPCL MSG :0          25 W. ROBINWOOD
ACCT CNTL:0          PALO ALTO, CA.  94303                           415 765-8764


TYPE   CUR BAL   30 DAY    60 DAY    90 DAY   120/OVER   YTD PAY     LP$        LPD

  1      0.00      0.00      0.00      0.00      0.00       0.00      0.00 00/00/00
  4    395.00      0.00      0.00      0.00      0.00     339.00    339.00 07/16/77

TYPE      POLICY #    DATA CODE     VALUE

  4     94-34343            A       HARRY'S HOISERY
                            C       394-43-4353
                            E       43

PT          PATIENT          SEX  BIRTHDATE  DATA CODE  VALUE

AM   ANTHONY MAXWELL          M     04/07/39      B       1
SM   SAMANTHA MAXWELL         F     11/25/42      B       1

CASE#  PT   TYPE  DR   HOLD   DIAG     DATE      CODE   #    SRF     AMOUNT    BALANCE

  2    AM     4    4    N        2  06/30/77     117                   9.00       9.00
                                    06/30/77     118                   1.00      10.00
                                    06/30/77      50                  15.00      25.00
                                    06/30/77       1                 200.00     225.00
                                    07/16/77     613    9    DL       24.00     249.00
                                    07/16/77     613   12    MOD      10.00     259.00
                                    07/16/77     230   23             25.00     284.00

  1    AM     4    2    N        2  06/30/77      20                 250.00     250.00
                                    06/30/77     250                  35.00     285.00
                                    06/30/77     512                 125.00     410.00
                                    06/30/77     671                  40.00     450.00
                                    07/16/77      16                -339.00     111.00
```

Fig. 11. Sample ledger showing all information stored on each account's computer file.

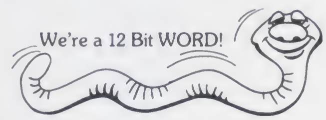
```
                 ACCOUNT NUMBERS/NAMES

                      JUL 23, 1977


ACCT#              ACCOUNT NAME
-----         -------------------------

10032         AHL, NEWETT
10020         ANDERSON, PAUL
10010         BLACKBURN, ARNOLD
10030         BUNKER, ARCHIE D
10011         CAVENDISH, FRANK
10008         DEBUNKER, ARCHIE
10009         FERNWOOD, FREDDIE
10005         GOLDFINGER, PHILLIP
10031         HOLMS, O.W.
10026         JACKSON, STEPHANIE
10012         JAMESON,ELLIE
10002         JOHNSON, TOM
10007         LOCKWOOD, ROBERT M
10006         MANCHESTER, CHARLES
10021         MARTIN, LOUISE
10023         MAXWELL, ANTHONY
10027         MCGRAW, ALLEN
10025         MEADOWS, JACK
10018         MELROSE, JONATHON
10022         NEWMAN, STEVEN
10001         REYNOLDS, JOHN L
10004         SHERMAN, MARY
10019         SHERMAN, WALTER
10016         SIMPLE, I. M.
10024         SUMMER, SCOTT
10029         WALTERS, MICHELL
10015         WATSON, JOHN M
10003         WHITTLER, ROBERT C
10028         ZENTNER, JOHN B
```

Fig. 10. Reports giving account number and name can be sorted alphabetically or by account number.

ITEM OF THE MONTH

FASTER THAN A SPEEDING

More Powerful Than Any 8 Bit BASIC



Able To Leap through a 600 line program in a single bound*!

def: single bound - 8 K Words of Memory TOTAL!

compare to your 8 bit machine

NO

IT'S

★ SUPER BASIC ★

by TLF

Proof is in the PUDDING IT IN YOUR MINI 12 AND TRYING IT!

SUPERBASIC COMMAND SUMMARY		FUNCTIONS		SUPERBASIC STATEMENT SUMMARY		
BYE	OVERLAY	ABS	SIN	CHAIN	INPUT LINE	OPEN FILE
CATALOG	PUNCH	ATN	SQR	CLOSE	KILL	PRINT
COMPILE	RENAME	COS	TAN	DATA	(LET)	PRINT USING
DELETE	REPLACE	EXP	POS	DIM	NEXT	RANDOMIZE
EDIT	RESEQUENCE	INT	TAB	END	ON..GOTO	READ
EXECUTE	RUN	LOG	ASCII	FOR	ON..GOSUB	REM
LENGTH	SAVE	PI	CHR $	GOSUB	ON EOF GOTO	RESTORE
LIST	SCRATCH	RND	LEN	GOTO	OPEN file for input	RETURN
MARGIN	SEARCH	SGN	MID	INPUT	OPEN file for output	STOP
NEW	TAPE			IF THEN line #	IF THEN statement	
OLD	UNSAVE					

T20

Write today for your FREE Software Summary from TLF.

P.S. You can get SUPERBASIC FREE with purchase of MINI 12 computer from

TLF Corporation P.O. Box 2298 Littleton Colorado 80161
Telephone 303 922 6241 Telex 454541

ACCOUNTS RECEIVABLE AGING

JUL 19, 1977

TYPE	ACCT#	NAME	LAST PAYMT	LAST PAY	PAY YTD	BALANCE	CURRENT	30 DAY	60 DAY	90 DAY	120/OVER
1	10020	ANDERSON, PAUL	08/03/77	25.00	25.00	23.00	0.00	23.00	0.00	0.00	0.00
1	10010	BLACKBURN, ARNOLD	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10011	CAVENDISH, FRANK	08/25/77	45.00	45.00	94.00	0.00	0.00	94.00	0.00	0.00
1	10005	GOLDFINGER, PHILLIP	07/05/77	500.00	500.00	3.00	0.00	0.00	3.00	0.00	0.00
1	10002	JOHNSON, TOM	07/05/77	300.00	300.00	75.00	0.00	0.00	0.00	0.00	75.00
1	10007	LOCKWOOD, ROBERT M	08/25/77	45.00	120.00	203.00	0.00	0.00	173.00	0.00	30.00
1	10006	MANCHESTER, CHARLES	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10021	MARTIN, LOUISE	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10023	MAXWELL, ANTHONY	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10025	MEADOWS, JACK	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10018	MELROSE, JONATHON	08/03/77	10.00	40.00	6.00	0.00	6.00	0.00	0.00	0.00
1	10022	NEWMAN, STEVEN	08/03/77	25.00	25.00	22.00	0.00	22.00	0.00	0.00	0.00
1	10001	REYNOLDS, JOHN L	07/05/77	150.00	150.00	100.00	0.00	0.00	0.00	0.00	100.00
1	10004	SHERMAN, MARY	08/13/77	5.00	185.00	140.00	8.00	0.00	132.00	0.00	0.00
1	10019	SHERMAN, WALTER	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10016	SIMPLE, I.M.	08/03/77	35.00	35.00	35.00	0.00	35.00	0.00	0.00	0.00
1	10024	SUMMER, SCOTT	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10015	WATSON, JOHN M	/ /	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	10003	WHITTLER, ROBERT C	07/05/77	37.50	37.50	0.00	0.00	0.00	0.00	0.00	0.00
		SUBTOTAL			1462.50	701.00	8.00	86.00	402.00	0.00	205.00
						100.0%	1.1%	12.2%	57.3%	0.0%	29.2%

Fig. 12. This report, which has a separate page for each payment type, shows all accounts, in alphabetical order, with the corresponding payment type. Each account's payments are shown, as are account age and amount. This report allows you to establish who is not paying promptly.

DELINQUENCY REPORT

JUL 24, 1977

TYPE	ACCT#	NAME & ADDRESS	TELEPHONE	BALANCE	DELINQUENT	LAST PAYMT	LAST PAY $	COMMENTS
1	10002	JOHNSON, TOM 876 LEMON MENLO PARK,CA 94025	415 854-5786	75.00	75.00	07/05/77	300.00	
7	10007	LOCKWOOD, ROBERT M 456 ALMA PALO ALTO,CA 94306	415 349-1729	659.00	562.00	08/25/77	70.00	
1	10007	LOCKWOOD, ROBERT M 456 ALMA PALO ALTO,CA 94306	415 349-1729	203.00	30.00	08/25/77	45.00	
5	10006	MANCHESTER, CHARLES 12896 EL CAMINO REAL CUPERTINO, CA 94523	408 349-9872	1250.00	825.00	07/25/77	50.00	
1	10001	REYNOLDS, JOHN L 1224 PINECREST WAY ATHERTON,CA 94025	415 345-7878	100.00	100.00	07/05/77	150.00	

Fig. 13. All accounts receivable with due dates over 120 days. This is an "action" report used to contact the overdue account and request payment. Pertinent account information is given and space is available for comments, should they be necessary.

support personnel, which services are generating the greatest revenue and hence should be emphasized more, and which services are taking a lot of time but generating little revenue (perhaps these should be referred to other doctors or delegated to assistants if possible).

Since two of the categories are payments and adjustments, the doctor can determine if any unauthorized write-offs were made during the month. He may also analyze the ways his receipts were collected to determine if insurance, cash or credit services are important factors in his practice. This last report is certainly useful and important since the doctor can keep track of how his practice is operating and who is doing the work. Since it is broken down according to doctor or hygienist, it is also a useful report for year-end performance review purposes.

In addition to the month-end reports, several month-end programs may optionally be run. These include conversion of past services and balances into a single balance-forward amount for each account starting at a given date. In addition, all accounts with zero balances can optionally be removed from the data base. Moreover, the usual month-end utilities such as clearing of monthly totals of receipts, charges, etc., is normally done after all reports are generated.

As a final backup procedure, the month-end disks are copied to a new set of disks, and these are stored together with a hard copy of the entire patient ledger. These two items are usually stored off-site in a safe-deposit box or something similar. Thus, the entire practice's monthly information is available, should any catastrophe happen in the office. As a benefit of this monthly backup system, the doctor can easily go back and review the history of services performed and charges billed simply by running the appropriate month-end disk in the computer.

After the month-end programs are run, the system is ready to resume the daily procedures described earlier. Both of these cycles continue until the end of the year. At this time, monthly reports are again run, except they show both month-to-date and year-to-date figures. Final bills and insurance forms are generated, and the year-end disks are saved off-site with the year-ending reports. A year-to-date clearing program is then run to adjust the files and set them up for the first month of the new year. The daily procedures then resume, and the cycle starts over with all old account and code information preserved from the previous year.

The complete software system just described allows a doctor to keep track of all patient accounts in a uniform manner; it keeps track of services performed, charges billed and receipts received; it also allows a doctor to measure the pulse of his prac-

tice much as he determines the health of a patient. And finally, it enables him to practice what he was taught rather than become drowned in a sea of paperwork, insurance red tape and accounting terminology and procedures that are better left to a noncomplaining computer!

Training and Education

Both the hardware and software described previously are often complex and difficult for the computer layman to understand. Consequently, Promedics has instituted a two-day training and education course for all users of the PAC 1 system. This session is meant to acquaint the new computer user with terminology, procedures and database management. The usual session includes the following topics:

1. General system configuration and machine operation.
2. Data input procedures.
3. Report generation.
4. Computer operations and procedures.

The training session is as important as reliable hardware and debugged software, since you can't do anything with the latter two unless you know how. Additional training is also provided when new users face exceptional problems or if a computer language is being taught to a programmer.

Maintenance

Not only is training important in order for someone to be able to run the software, but reliable equipment is also a prerequisite to providing turnkey systems to professionals. The PAC 1 system uses industry-proven components chosen not for least cost but for cost effectiveness — that is, a minimal cost from a purchase and a maintenance standpoint.

As an example, it makes little sense to put a $160, 8K RAM board in a computer if it has inferior-quality ICs or bad temperature stability, since erratic behavior in either the hardware or software is likely to occur. It makes much more sense to purchase an 8K memory for $265 that comes with a one-year guarantee and is burned in and tested as a working unit using software to be used with the unit! It may cost a bit more to take these precautions, but when a doctor's financial lifeline is on the computer, you don't want to use cheap components that aren't reliable.

In addition to reliable components, a user must have a system that is working at

ANALYSIS OF SERVICES: TOTALS

AUG 30, 1977

SVCOD	DESCRIPTION	CURRENT MONTH TRANSACTIONS					YEAR-TO-DATE TRANSACTIONS				
		NUMBER	% TTL#	AVE AMT	TOTAL AMT	% TTL$	NUMBER	% TTL#	AVE AMT	TOTAL AMT	% TTL$
MISC SERVICES											
TOTAL		0	100.00	0.00	0.00	100.00	0	100.00	0.00	0.00	100.00
DIAGNOSTIC											
20	OFFICE VISIT	12	35.29	8.00	96.00	19.23	16	28.57	8.00	128.00	12.96
30	PROFESSIONAL VISITS	0	0.00	0.00	0.00	0.00	1	1.78	15.00	15.00	1.51
40	SPECIAL CONSULTATION	2	5.88	15.00	30.00	6.01	2	3.57	15.00	30.00	3.03
50	PROPHYLAXIS-ADULT	7	20.58	15.00	105.00	21.04	9	16.07	15.00	135.00	13.67
62	FLUORIDE TREATMENT	2	5.88	15.00	30.00	6.01	5	8.92	15.00	75.00	7.59
112	DENTURE SERIES FILMS	1	2.94	22.00	22.00	4.40	3	5.35	22.00	66.00	6.68
116	BITE WINGS & EXAM	0	0.00	0.00	0.00	0.00	1	1.78	7.00	7.00	.70
117	BITE WINGS-4 FILMS	5	14.70	9.00	45.00	9.01	6	10.71	9.00	54.00	5.47
118	BITE WINGS, ADDITIONAL FILM	1	2.94	1.00	1.00	.20	2	3.57	1.00	2.00	.20
125	PANOREX FILM	2	5.88	25.00	50.00	10.02	4	7.14	25.00	100.00	10.13
160	MICROSCOPIC EXAM	0	0.00	0.00	0.00	0.00	1	1.78	15.00	15.00	1.51
666	ORTHO EXAM	2	5.88	60.00	120.00	24.04	6	10.71	60.00	360.00	36.47
TOTAL		34	100.00	14.67	499.00	100.00	56	100.00	17.62	987.00	100.00
ORAL SURGERY											
200	SINGLE EXTRACTION	3	42.85	12.00	36.00	20.45	4	33.33	12.00	48.00	10.59
201	ADDITIONAL EXTRACTION	1	14.28	10.00	10.00	5.68	2	16.66	17.50	35.00	7.72
231	REMOVAL OF TOOTH (2)	1	14.28	45.00	45.00	25.56	1	8.33	45.00	45.00	9.93
232	REMOVAL OF TOOTH (3)	0	0.00	0.00	0.00	0.00	1	8.33	55.00	55.00	12.14
250	ALVEOLECTOMY	1	14.28	35.00	35.00	19.88	1	8.33	35.00	35.00	7.72
280	EXCISION OF CYST	0	0.00	0.00	0.00	0.00	1	8.33	35.00	35.00	7.72
281	EXCISION OF LARGE CYST	1	14.28	50.00	50.00	28.40	1	8.33	50.00	50.00	11.03
400	ANETHESIA	0	0.00	0.00	0.00	0.00	1	8.33	150.00	150.00	33.11
TOTAL		7	100.00	25.14	176.00	100.00	12	100.00	37.75	453.00	100.00
PERIDONTICS											
451	EMERGENCY TREATMENT	0	0.00	0.00	0.00	0.00	1	16.66	15.00	15.00	4.28
452	SUBGINGIVAL CURETTAGE	1	33.33	20.00	20.00	9.09	1	16.66	20.00	20.00	5.71
453	CORRECTION OF OCCLUSION	0	0.00	0.00	0.00	0.00	1	16.66	15.00	15.00	4.28
472	GINGIVECTOMY	2	66.66	100.00	200.00	90.90	3	50.00	100.00	300.00	85.71
TOTAL		3	100.00	73.33	220.00	100.00	6	100.00	58.33	350.00	100.00
ENDODONTICS											
502	VITAL PULPOTOMY	0	0.00	0.00	0.00	0.00	1	33.33	12.00	12.00	8.16
503	RECALCIFICATION	1	50.00	10.00	10.00	7.40	1	33.33	10.00	10.00	6.80
512	BI-ROOTED CANAL	1	50.00	125.00	125.00	92.59	1	33.33	125.00	125.00	85.03
TOTAL		2	100.00	67.50	135.00	100.00	3	100.00	49.00	147.00	100.00

RESTORATIVE

600	ONE SURFACE AMALGAM	0	0.00	0.00	0.00	0.00	5	17.85	10.00	50.00	7.23
601	TWO SURFACE AMALGAM	1	7.69	15.00	15.00	3.44	2	7.14	15.00	30.00	4.34
602	THREE SURFACE AMALGAM	1	7.69	20.00	20.00	4.59	6	21.42	20.00	120.00	17.36
603	FOUR SURFACE AMALGAM	1	7.69	25.00	25.00	5.74	2	7.14	25.00	50.00	7.23
611	ONE SURFACE RESTORATION	3	23.07	12.00	36.00	8.27	3	10.71	12.00	36.00	5.20
612	TWO SURFACE RESTORATION	2	15.38	18.00	36.00	8.27	3	10.71	18.00	54.00	7.81
613	THREE SURFACE RESTORATION	2	15.38	24.00	48.00	11.03	4	14.28	24.00	96.00	13.89
636	TWO SURFACE GOLD	1	7.69	100.00	100.00	22.98	1	3.57	100.00	100.00	14.47
637	THREE SURFACE GOLD	1	7.69	130.00	130.00	29.88	1	3.57	130.00	130.00	18.81
638	GOLD ONLAY	1	7.69	25.00	25.00	5.74	1	3.57	25.00	25.00	3.61
TOTAL		13	100.00	33.46	435.00	100.00	28	100.00	24.67	691.00	100.00

CROWN & BRIDGE

652	PORCELAIN CROWN	0	0.00	0.00	0.00	0.00	1	12.50	150.00	150.00	21.42
660	GOLD CROWN	1	33.33	160.00	160.00	75.47	3	37.50	160.00	480.00	68.57
671	SS CROWN (PERM)	1	33.33	40.00	40.00	18.86	1	12.50	40.00	40.00	5.71
685	INLAY	0	0.00	0.00	0.00	0.00	1	12.50	6.00	6.00	.85
687	BRIDGE REPAIRS	1	33.33	12.00	12.00	5.66	2	25.00	12.00	24.00	3.42
TOTAL		3	100.00	70.66	212.00	100.00	8	100.00	87.50	700.00	100.00

PROSTHETICS

701	COMPLETE MANDIBULAR DENTURE	1	50.00	250.00	250.00	80.64	1	25.00	250.00	250.00	64.10
705	SIMPLE STRESS BREAKERS	0	0.00	0.00	0.00	0.00	2	50.00	40.00	80.00	20.51
706	STAYPLATE	1	50.00	60.00	60.00	19.35	1	25.00	60.00	60.00	15.38
TOTAL		2	100.00	155.00	310.00	100.00	4	100.00	97.50	390.00	100.00

ORTHODONTICS

800	FIXED SPACE MAINTAINER	1	33.33	75.00	75.00	78.94	2	28.57	75.00	150.00	57.69
801	SS SPACE MAINTAINER	0	0.00	0.00	0.00	0.00	1	14.28	75.00	75.00	28.84
802	STAINLESS STEEL CLASPS	0	0.00	0.00	0.00	0.00	1	14.28	5.00	5.00	1.92
803	STUDY MODELS	2	66.66	10.00	20.00	21.05	3	42.85	10.00	30.00	11.53
TOTAL		3	100.00	31.66	95.00	100.00	7	100.00	37.14	260.00	100.00

BROKEN APPOINTMENTS

999	BROKEN APPOINTMENT	0	0.00	0.00	0.00	0.00	1	100.00	25.00	25.00	100.00
TOTAL		0	100.00	0.00	0.00	100.00	1	100.00	25.00	25.00	100.00

BALANCE FORWARD

1	BALANCE FORWARD	0	0.00	0.00	0.00	0.00	11	100.00	410.22	4512.50	100.00
TOTAL		0	100.00	0.00	0.00	100.00	11	100.00	410.22	4512.50	100.00

ADJUSTMENTS

2	ERROR -- OVERCHARGE	0	0.00	0.00	0.00	0.00	1	20.00	-12.50	-12.50	34.72
3	ERROR -- UNDERCHARGE	0	0.00	0.00	0.00	0.00	1	20.00	25.00	25.00	-69.44
4	ERROR -- OVER PAYMENT	1	33.33	-10.00	-10.00	20.61	1	20.00	-10.00	-10.00	27.77
9	INSURANCE CREDIT	1	33.33	-25.00	-25.00	51.54	1	20.00	-25.00	-25.00	69.44
10	OTHER ADJUSTMENT	1	33.33	-13.50	-13.50	27.83	1	20.00	-13.50	-13.50	37.50
TOTAL		3	100.00	-16.16	-48.50	100.00	5	100.00	-7.20	-36.00	100.00

PAYMENTS

11	CASH PAYMENT	1	4.54	10.00	10.00	1.04	4	9.75	124.37	497.50	11.78
12	CHECK PAYMENT	9	40.90	28.33	255.00	26.69	14	34.14	72.14	1010.00	23.92
13	C.D.S. PAYMENT	2	9.09	50.00	100.00	10.46	4	9.75	176.25	705.00	16.70
14	MEDI-CAL PAYMENT	2	9.09	72.65	145.30	15.20	4	9.75	205.07	820.30	19.43
15	WELFARE PAYMENT	2	9.09	60.00	120.00	12.56	5	12.19	134.60	673.00	15.94
16	MEDICAID PAYMENT	3	13.63	63.33	190.00	19.88	4	9.75	61.25	245.00	5.80
17	MISC INSURANCE PAYMENT	3	13.63	45.00	135.00	14.13	5	12.19	53.00	265.00	6.27
18	OTHER PAYMENT	0	0.00	0.00	0.00	0.00	1	2.43	5.00	5.00	.11
TOTAL		22	100.00	43.42	955.30	100.00	41	100.00	102.94	4220.80	100.00

This report gives an analysis of services by doctor, and total for the practice broken down by service category and service type. Both number of services and amount of services are given in conjunction with percentages. This report is useful for determining the amount of services provided and the average amounts received. A careful analysis of this report may enable you to determine which services can be expanded, given to your hygienist, or reviewed for possible rate change.

Fig. 14. An analysis of services by doctor, and total for the practice broken down according to service category and service type. Both number of services and amount of services are given in conjunction with percentages. This report is useful for determining which services can be expanded, given to your hygienist or reviewed for possible rate change.

least 95 percent of the time. This means service has to be provided in a given amount of time, usually several hours to a day depending upon the importance of downtime. The PAC 1 is serviced both by Promedics and a national service organization, thus ensuring minimum downtime and maximum computer usage. A maintenance plan is important and often necessary if you lease a computer. Make sure you have one — it pays for itself in minimizing expensive downtime (like during a month-end statement run).

System Expansion

In considering the virtues of different computers, it is important to think about system expansion both from a hardware and software point of view. The PAC 1 system is upgradable in both areas.

From a hardware standpoint, the system can accommodate additional memory for more sophisticated programs, increased disk drive capacity for greater on-line data storage of patient information, and additional terminals at faster speeds for faster processing of statements, insurance forms and reports. The hardware can also be configured to communicate with larger computers such as the IBM 370, Burroughs 6700, etc., and then act as a remote terminal that transmits, receives and stores information.

Additional software can also be accommodated. A soon-to-be released multiuser system will expand the capabilities of the PAC 1 so that several jobs may be running at the same time. A good example would be the simultaneous running of the patient statements and the daily edit program, where data for new accounts is being input. Also, additional software such as accounts payable, general ledger, payroll and inventory control can easily be accommodated to perform more of the accounting control tasks in an office or small business. These programs will be available to PAC 1 users (and others) simply by insertion of a different disk with new programs and data files. Furthermore, each will run as an independent module, or all can be combined as an integrated system to generate income statements and balance sheets.

The capability to add further software and hardware enables the user to maintain a flexible position by allowing the computer to increase its taskload as the business grows.

System Costs

A complete turnkey computer system as described, including delivery and installation, training, a supply of insurance forms and patient statements, operating supplies and a full year of software support, retails for $12,000 to $15,000, depending on the hardware and software options selected. Several lease plans are available that bring the cost down to less than the price of hiring an additional assistant.

Both the hardware and the software are available separately as is the high-level language compiler/interpreter. Further information on this language may be obtained from Promedics.

Conclusion

The day of the turnkey microcomputer system is here, and machines and software are already in the field performing tasks similar to computers costing thousands of dollars more. When considering a system for your practice or business, choose one that is easy to use, comes complete with everything you need, features a thorough training and education session and can be expanded with both additional software and hardware. ■

Rich Man, Poor Man, Merchant, Physician, Teacher, Lawyer, Student, Musician....

There's an iCOM Floppy Disk System for Everyone!

More Uses

People from every walk of life are adding iCOM® Floppy Disks to their microcomputers for such diverse tasks as payroll, inventory control, mailing lists, game playing, record keeping, parts ordering . . .

We've uncovered some innovative applications, too: The sailboat architect who puts equations and algorithms on an iCOM disk to test his nautical theories; the student who has automated a bowling alley; the iCOM dealer who designed an environmental control system for a university.

More Speed

These users have found iCOM floppies to be much faster and more versatile than cassette or paper tape. With iCOM, programs can be loaded in seconds; files updated in minutes; hundreds of programs can be stored on a single disk.

More Models

iCOM has Frugal Floppies™, Dual Floppies, Microfloppies™ (using the new 5 ¼'' diskette), and other new approaches to floppy disk systems. Each is hardware and software compatible with Altair™, IMSAI, Poly 88, Sol-20 and other microcomputers using the Altair S-100 bus format.

More Software

Then there's iCOM's famous software: Powerful field-proven FDOS-II with macro-assembler, string-oriented text editor, and file manager. Plus easy-to-use compatible 8K Disk BASIC. Each with super features such as: named variable length files, auto-file create, open and close, multiple merge and delete . . . and more.

More Backup

We've been building floppies for microcomputers for more than 3 years. Long before the rest. Thousands of systems are operating perfectly in the field. And we're part of Pertec Computer Corporation, one of the largest manufacturers of peripherals, microsystems, data entry products and data processing systems. We'll be around whenever you need us.

More Dealers

Maybe not in quantity, but in quality. We've chosen our dealer network carefully to assure you of assistance every step of the way. Our prices are right. Our delivery is fast. Our dealers are experienced and knowledgeable.

Must Reading

Our free booklet, ''What a Floppy Disk Can Do for You'' is must reading. Send for yours today or visit your dealer.

icom® MICROPERIPHERALS®

a division of Pertec Computer Corporation

6741 Variel Ave., Canoga Park, CA 91303 U.S.A.
Tel. (213) 348-1391 TWX 910-494-2788

John Eaton
435 W. Padre #W 10
Santa Barbara CA 93105

Growing with KIM

expansion PC board

The KIM-1 Microcomputer by MOS Technology is a versatile computer-on-a-card that is used by many hobbyists. However, a disadvantage of the KIM-1 is its incompatibility with the Altair bus. This article describes the construction and operation of an interface board that allows the KIM-1 owner to use a variety of peripherals available for the Altair bus.

How Does It Work?

The schematic for the KIM Expander board is shown in Fig. 1. Connector pins to the rest of the system are labeled as follows: KE, KIM-1 Expansion connector; KA, KIM-1 Applications connector; and S, for the Altair bus pin number. The 16 address lines from the KIM-1 are buffered by ICs 3, 4, and 5 to drive the Altair bus address lines. The three highest address lines (A13, A14, and A15) also drive a 3/8 decoder (IC6). The 0 and 7 outputs of IC6 are connected together and control the KIM-1 decode enable line (KA-K). If the KIM-1 addresses the normal KIM range of 0000_h-$1FFF_h$, then the KIM-1 on-board decoder will be enabled by the 0 output of IC6. If the KIM addresses in the range of $E000_h$-$FFFF_h$, then the 7 output of IC6 will enable the KIM-1. This means that the RESET, NMI, and IRQ vectors that are stored in ROM on the KIM-1 will be properly fetched by the CPU.

Close-up of KIM and Expander board.

The KIM-1 has an 8-bit bidirectional data bus for data transfers, while the Altair bus uses an 8-bit Data Out bus and a separate 8-bit Data In bus. ICs 1 and 2 perform the function of buffering and splitting the KIM-1 data bus.

Care must be taken to ensure that the Data In buffers are enabled only when the CPU is reading data from the Altair bus. If the Data In buffers are enabled while the CPU is addressing the KIM-1 on-board memory, the data would be lost or altered. In

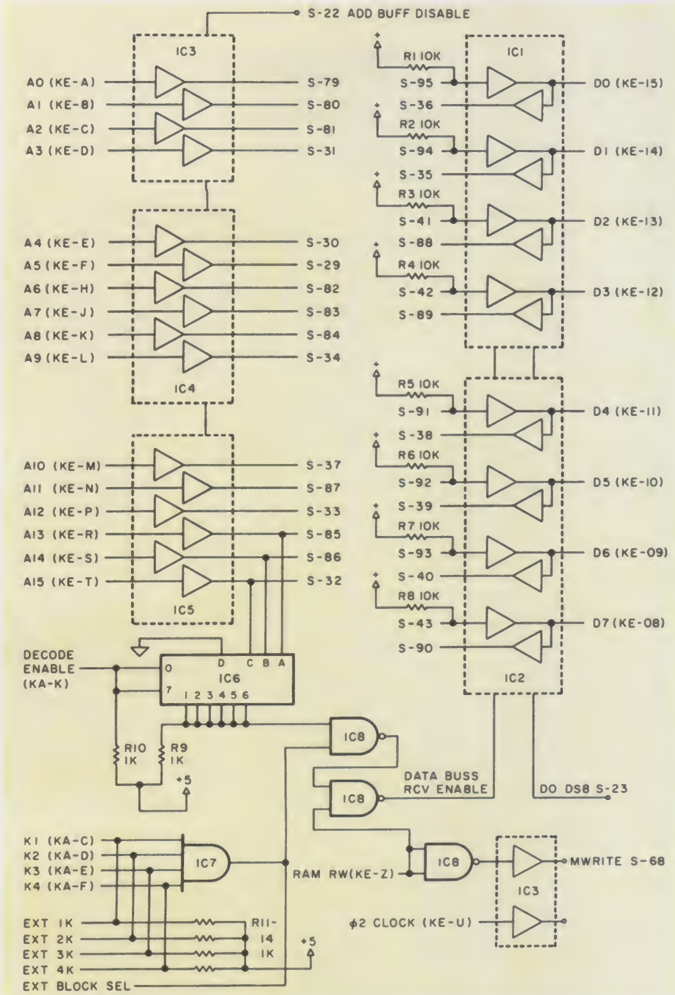


Fig. 1a. Schematic diagram of KIM Expander board.

order to ensure that only one data source is enabled at any one time a special control signal is generated. IC6, which supplied the KIM-1 decode enable signal, is used to determine when the Altair bus is being addressed. Outputs 1 through 6 are wired together and are normally high but will drop to 0 when the CPU addresses in the range of 2000_h-$DFFF_h$. Most of the memory and peripherals plugged into the Altair bus should reside in this range.

The only exception to this is a 4K memory block that is left vacant by the KIM-1. The KIM-1 resides in an 8K block of memory but only uses 4K for memory and I/O. The unused 4K is divided into four 1K blocks by the decoder on the KIM-1 board and is available on the KIM-1 applications connector.

In order to allow the user to fill this void with a memory board, the KIM Expander ANDs these four 1K outputs (K1, K2, K3 and K4) together with IC7. The result is a normally high signal that drops to 0 when the CPU addresses from 0400_h to $13FF_h$.

This signal is NANDed by IC8 with the Altair bus selected signal from IC6. The result is a signal that is normally low but rises to logic 1 whenever the CPU is addressing memory that is on the Altair bus. This signal is then gated with the RAM R/W signal from the KIM-1 to disable the Data In buffers when the CPU is either outputting data or reading from on-board memory.

The RAM R/W signal is also inverted and sent to the Altair bus via IC3. This signal acts as a MWRITE signal for the bus and goes from 0 to 1 during Ø2 of a write cycle. This is the only time that a MOS 6502 is outputting valid data. The Ø2 clock is buffered and available if any of the user's peripherals require a systems clock signal.

In order to properly use the KIM Expander some jumpers must be placed on the Altair bus. ADDRS DIS (S-22) and Data Out DIS (S-23) must be jumpered to ground. These should only be disconnected to implement a second processor or a DMA board. Raising these lines above ground will effectively disable the buffers and remove the address, Data out, MWRITE and Ø2 clock signals from the Altair bus. Control of the bus can then be assumed by another board. The user may want the KIM-1 to operate using its on-board memory while the bus is being controlled by another device. If so, the KIM-1 Decode Enable should be grounded during the time that the Address buffers are disabled.

The Altair bus was designed for an 8080 microprocessor, which uses a different type of input/output than a MOS 6502. Three lines on the bus, (S_{memr}, S_{in}, and S_{out}) are used to tell the system whether the processor wants memory, an input channel or an output channel. Since the MOS 6502 does not have input/output instructions, it handles everything as a transfer to or from a memory location. Therefore, S_{in} (S-46) and S_{out} (S-45) must be jumpered to ground and S_{memr} (S-47) must be jumpered to +5 V dc in order to operate most memory boards. PDBIN (S-78) should also be jumpered to +5 V dc.

Compatibility

Although the KIM Expander will allow the KIM-1 to control many Altair compatible boards, it will not control all of them. There are some basic differences between a MOS 6502 and an Intel 8080 that cannot be compensated for by the KIM Expander.

An example of this is the Cromemco Bytesaver board. This ROM reader/programmer is designed to program ROMs by having the CPU write data into the desired ROM location. This kicks off an internal timing chain on the board that will stop the processor via the READY line. It also pulses the write enable pins on the

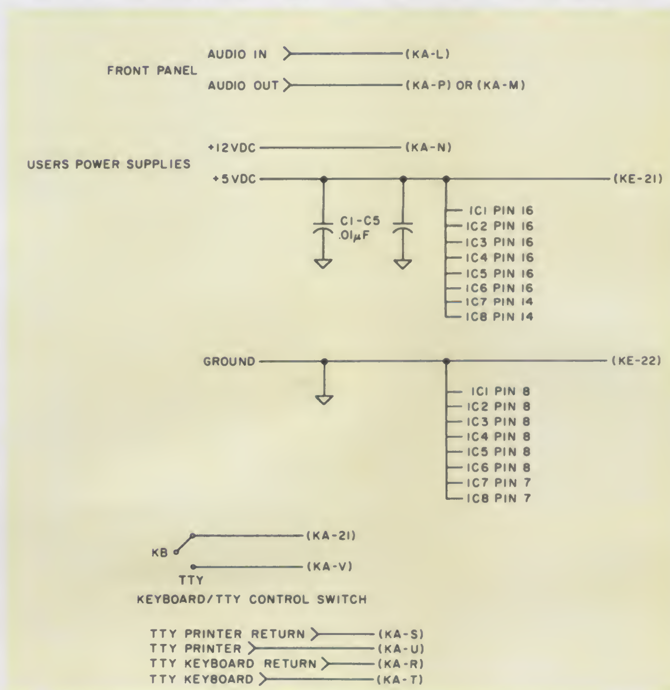


Fig. 1b. Schematic diagram of power distribution and I/O connections.

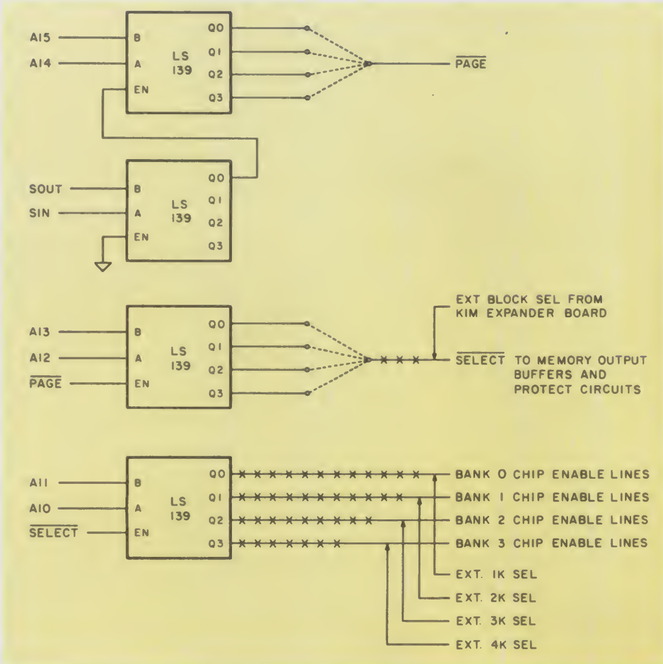


Fig. 2. Modifications on Econoram 4K memory board to allow it to reside in KIM-1 memory space.

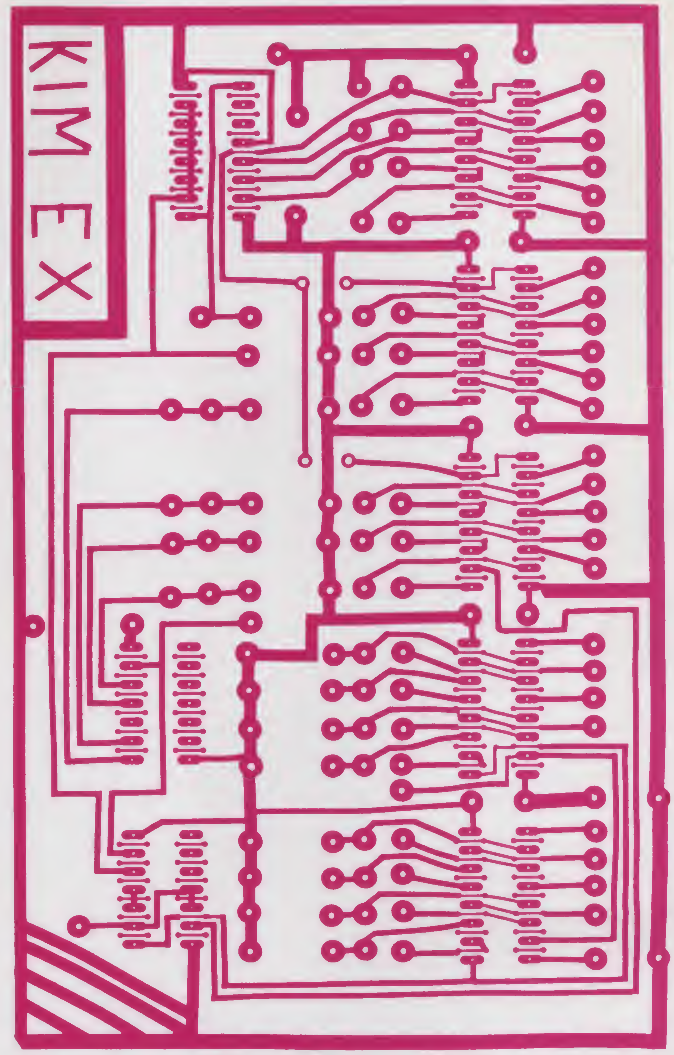


Fig. 3. Printed circuit diagram for the KIM Expander (full size).

ROM to allow it to accept the data that the CPU is holding on the Data Out bus. This works fine on an 8080-based system but will not work on a 6502-based system like the KIM-1. The reason is that a 6502 will not stop on a write cycle, and so the ROM address and data will not remain on the bus long enough to be accepted by the ROM. The KIM-1 will be able to read preprogrammed ROMs from the Bytesaver but will not be able to program them as an 8080 system can.

When considering buying any peripheral for use with this system, make sure that it will work with a MOS 6502. Some boards may require special signals that are not available from the KIM-1. Be especially careful about requirements for the ready line and clock signals. Memory boards that require the computer to wait are generally not compatible with the KIM-1; it will not hold data on the Data Out bus long enough for the memory to pick it up.

Memory Assignments

Filling the 4K memory void between 0400_h and $13FF_h$ requires special handling. Most memory boards can occupy only one of 16 preset slots in memory. This means that a 4K memory board can occupy from address 0000_h to $0FFF_h$ or from 1000_h to $1FFF_h$, but it could not fit in the area left empty by the KIM-1. One method of filling this void using a 4K memory board is shown in Fig. 2. This shows how a Godbout 4K Econoram can be placed in the KIM-1 system starting at address 0400. The four-chip enable lines to all the 2102s are disconnected from their on-board decoders and are reconnected to the K1 through K4 signals from the KIM-1. The EXT Block SEL signal from IC7 enables the Data Out buffers on the memory board whenever the KIM-1 addresses any memory in this 4K block.

If you use any memory board that has memory protect features then you may want to wire up a front panel switch to set or reset the protect status flip-flop on the board. A panel-mounted LED used as a status indicator also would be helpful. If this is not desired, then simply jumper the protect input (S-70) to ground. This assures that all memory will be unprotected.

Construction

The printed circuit diagram shown in Fig. 3 will easily fit on a 6 x 4 in. printed circuit board with enough space left at the corners to allow for mounting holes. A Vector CU 65/45-1R single-sided PC board was used in the prototype. Although connection to the Altair bus can be directly wired, it is recommended that a multipin connector be used to allow the KIM-1 and KIM Expander to be disconnected from the bus.

In my prototype system the power supply and a 22-slot motherboard are mounted in a 17x18x7 in. cabinet. The KIM-1 and the KIM Expander card are mounted in a 17x12x2 in. enclosure that mounts on top of the motherboard chassis. All connections between the two sections are made through two 37-pin D-type connectors. This provides for 74 pins — enough for all connections to the Altair bus plus all of the KIM-1 I/O pins. The KIM-1 is mounted to the top of the chassis using standoffs, and an access hole allows the keyboard and display to be used.

The Altair bus requires power supplies of +16, -16 and +8 V dc in order to operate the peripheral boards. The KIM-1 and KIM Expander require +5 and +12 V dc. When the system is assembled the KIM-1 should be operational. If pushing the

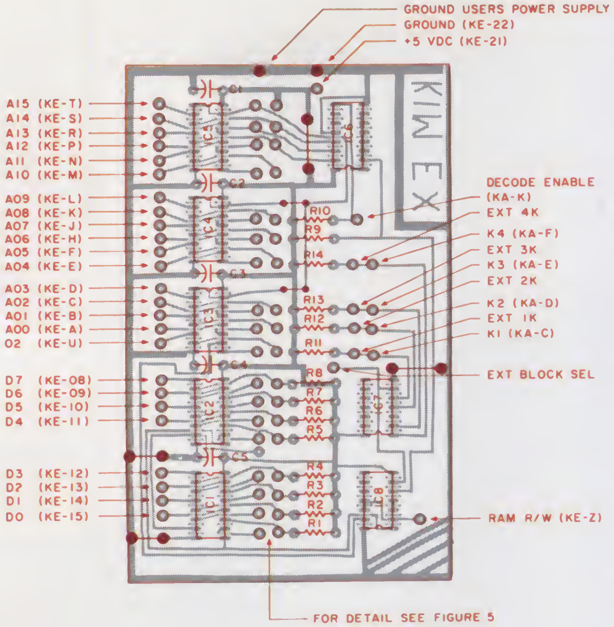


Fig. 4. Parts-placement and wiring diagram for connections on the KIM Expander Board.

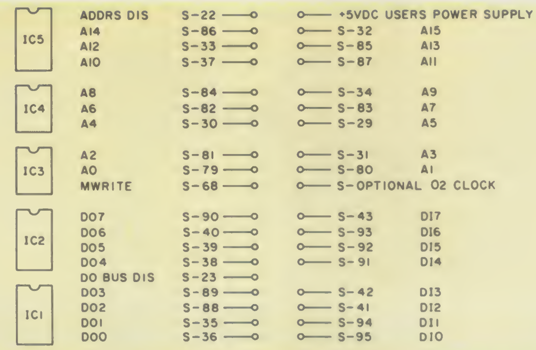


Fig. 5. Enlarged diagram of Altair bus connectors to the KIM Expander Board.

reset button does not cause the display to light try shorting the KIM-1 Decode Enable (KA-K) to ground. If it then works properly check IC6 for proper operation. If the display still remains dark, then remove ICs 1 and 2. If they are defective or not being enabled properly, the KIM-1 monitor program will not run.

Checkout and Troubleshooting

When you get the KIM-1 display lit you can use the keyboard to address any byte of memory on the Altair bus as well as the KIM-1 on-board memory. If there is no peripheral located at the selected address on the bus, the KIM-1 will display data as FF_h. This is different from the basic KIM-1, where an unimplemented address will cause its high order address byte to be displayed as data. You can verify this by addressing 0400_h with a basic KIM-1, and the data will read 04_h. With the KIM Expander and no external memory on the Altair bus it will read FF_h. When memory is placed on the bus the KIM-1 will be able to read and write into it in the same manner as its on-board memory.

You can troubleshoot the offboard peripherals using the KIM-1 keyboard and display if you understand how the KIM-1 monitor program operates. If you don't, even the simplest problem will seem monumental. For example, let's assume that you have a memory board with a bad chip in the Data 0 slot that causes the lowest bit in memory to always read 1. Testing this type of fault using an Altair-style front panel would be easy since you could write all 0s into memory and see the bad bit.

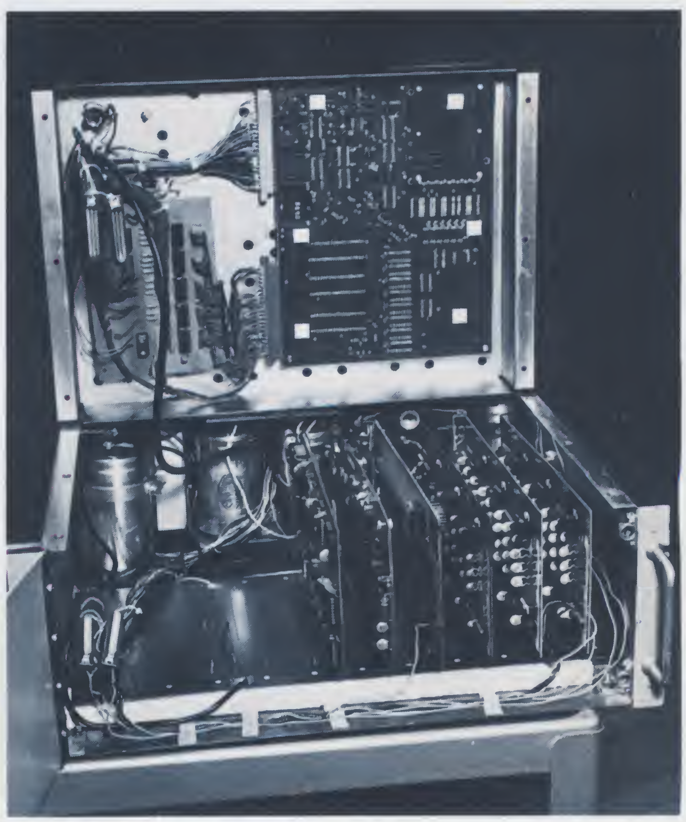
The KIM-1 uses software to perform all front panel operations. When data is to be deposited into memory, the old data is read out and the new data is serially shifted into a register one bit at a time, which is then written into memory. If any bits of memory are stuck at 1 or 0 the KIM-1 will just shift that value into the higher bits. In our example of data line 0 being stuck at 1, the result of trying to enter any data would be to set bits 1 through 7 to 1, and you would see FF_h on the display.

If the keyboard and display do not help in isolating the trouble, then try writing a short program to write a test value (00_h or FF_h) into memory and read it back.

By building the KIM Expander you can increase the power and versatility of the KIM-1. And you can use Altair compatible kits, which are normally less expensive than assembled units. ■

C 1-5	.01 uF bypass capacitor
IC 1&2	833 quad bus transceiver
IC 3,4&5	8097 hex bus driver
IC6	7415 BDC decimal decoder
IC7	7421 dual quad input AND
IC8	7400 Quad NAND gate
R 1-8	10k ¼ Watt resistor
R 9-14	1k ¼ Watt resistor

Table 1. Parts list.



The system. KIM-1 is mounted in top of cabinet along with Expander board. Motherboard with Altair bus boards is located in bottom.

REX™ IS COMING TO A COMPUTER STORE NEAR YOU!

RCC'S
REX™
The Total Computer System

IN TIME FOR CHRISTMAS

We took the most desirable small computer features and put them into one machine — **for $2495 (or less)!**

REX™, RCC's new personal computer, is space-age modern in appearance, regal in features, but down-to-earth in price.

REGAL FEATURES

Just add a hard copy terminal or our video display, select FORTRAN IV or Extended BASIC . . . or both, and begin developing your programs.

REX includes an S-100 motherboard with five card slots, 24K of RAM, Z-80 CPU, video display interface, bootstrap PROM and powerfail and vectored interrupt circuitry.

REX includes a modern walnut-panel cabinet, ASCII keyboard, 15 amp power supply, microfloppy drive (double-sided recording) and interface (including 8 bit parallel I/O port). All standard.

SOFTWARE TO PLEASE A KING

REX includes a powerful system Monitor, loaded from PROM. Low cost software options include ANSI FORTRAN IV, Extended Disk BASIC, Floppy Disk Operating System, File Management System, Text Editor and Linking Loader.

A KING'S CHOICE OF OPTIONS

Black and White or Color Video Monitors ■ Programmable CRT Character Generator ■ Expansion Microfloppy Drives ■ RAM Memory to 64K (does not require bus slots) ■ 120 CPS 80/96 Column Printer ■ RF Modulator and Color TV Driver for Home Sets ■ Programmable Serial Channel ■ Data Communications Modem.

ROYAL QUALITY AT A SURPRISINGLY LOW PRICE

The **REX** System is shipped assembled and tested for **$2495**, list price. See your computer dealer for his special low introductory price (for short time only).

DEALERS

CALIFORNIA
San Francisco . . . . Computer Store of San Francisco
San Jose . . . . Byte Shope III
Santa Clara. . . . . Byte Shop-Computer Store

COLORADO
Englewood. . . . . Byte Shop, Inc.
Boulder. . . . . The Byte Shop

IOWA
Davenport . . . . Computer Store of Davenport

MICHIGAN
Royal Oak . . . . Computer Mart of Royal Oak

MINNESOTA
Minneapolis . . . . Computer Depot

NEBRASKA
Lincoln. . . . . Microtech
Omaha . . . . Omaha Computer Store

NEW YORK
New York City . . . . Computer Mart of New York, Inc.

WISCONSIN
Beloit. . . . . Austin Computers

rcc™ **realistic controls corporation**

404 West 35th Street Davenport, IA 52806
Call Toll Free (800) 553-1863 In Iowa (319) 386-4400

Tandy COMPUTERS

A DIVISION OF TANDY CORPORATION

ANNOUNCING

GRAND OPENING

MOST COMPLETE LINE OF MICROCOMPUTERS AND ACCESSORIES

PLACE YOUR ORDER TODAY — CALL TOLL-FREE — 800-433-1679

RADIO SHACK

SOUTHWEST TECHNICAL PRODUCTS

CPU's — Alpha Micro, ICOM, IMSAI, Polymorphics, Processor Technology, T.D.L., Vector Graphics, Equinox

"POPULAR BRANDS" CARRIED

DISCS — ICOM, Digital Systems, IMSAI, Micropolis, North Star, Polymorphics, Processor Technology, Shugart, Smoke Signal, SWTP, Persci, Extensys

VIDEO BRAIN

TERMINALS — Beehive, Lear Siegler, SOROC, SWTP, Informer

PRINTERS — Centronics, Okidata, Practical Automation, SWTP, Diablo, IMSAI

Complete Line of Parts and Accessories!
Call for Special Prices! 800-433-1679

Mail To: TANDY COMPUTERS
P.O. Box 2936, Fort Worth, Texas 76102

Please Send Me TANDY COMPUTERS Full-Line 1978 Catalog

PLEASE PRINT

Name__________ Apt. _____

Street __________

City__________ State _____ Zip _____

Mark Garetz
HUH Electronic Music Productions
PO Box 259
Fairfax CA 94930

Has Godbout Done It Again?

Econoram II: $150 assembled

I'm sure most of you are aware that memory prices are starting to fall drastically. New and less expensive memory boards are introduced almost every day. A good percentage of these are quite dense (16K or higher) and are, in general, *dynamic* boards. This means they use dynamic memory chips that need their memory "refreshed" every couple of milliseconds. Making sure this happens correctly is no easy trick, especially in Altair bus systems. Since this technology is relatively new, no one seems to have made a dynamic board that works with everything, although the new Cromemco 16K dynamic board is rumored to be the best so far.

Consequently, the other major percentage of new memory boards use the proven *static* memory technology — chips that require no refresh. The most popular chip by far is the 2102 type (91L02, 21L02, 2102L1, etc., are all pin-compatible but they all have small differences such as speed, power dissipation, etc.), and the most popular board configuration for these chips is 8K. This is because it's physically impossible to get more than 64 2102s on a standard S100-size board.

But since the emphasis on these new boards is their lower cost, certain features are deleted on most of the boards as a consequence. For example, one board will not be fully buffered but will have plenty of bypass capacitors. Another will be fully buffered but have not a single bypass capacitor. Most have no wait states selectable. Some have addressing limitations. Some have no protect options. The list goes on, but I think you get the picture. To bring down the cost, one or more features are absent.

This article is, therefore, a review of what I consider to be the most outstanding value in the low-cost memory field. A board that has full features and some extra goodies on top of that. It's the new 8K Econoram II by Bill Godbout Electronics. Weighing in at $150 assembled, this board is unquestionably the winner. Let me explain why.

Proven Static Technology

This article will not even attempt to argue the pros and cons of static vs. dynamic memory. Suffice it to say that at this point I just don't trust dynamics — at least in Altair-bus-type systems.

The Econoram II uses the proven 2102L1 memory chip. It is a low-power, extra-fast version of the standard 2101. The chip that Bill uses has a guaranteed access time of 350 ns. They are all new, prime parts, despite Godbout Electronics' being a famous surplus house.

Econoram Board Features

Fully buffered: When a manufacturer tells you that his board is "fully buffered," make sure you read the fine print. That sometimes means that only the address lines are fully buffered, or only the data out lines are buffered.

The Econoram II is truly fully buffered. This means everything. "What is the advantage of that?" I hear you ask. Well, without going into great detail (it would take a whole article, and, anyway, I'm not sure I'm qualified to write it), the processor board has "bus drivers" on it that are usually capable of driving quite a few "loads." A static memory chip doesn't present much of a load to the bus all by itself, but there are 64 of them on each 8K board. Everything is fine for the first couple of boards, but along about the third one, the bus starts looking pretty terrible, and although it might work, there's not much margin for error left. In other words, things start to run pretty close to the edge. Then you plug in the fourth board and things just don't work the same anymore.

I'm not trying to scare you (well, scare maybe, but not terrify). The point of all this is that if you have a buffer for each card, that buffer can handle its 64 loads OK, and all the processor card's buffers have to worry about are the loads presented by the memory board buffers — not all 64 of the chips!

Unique addressing options: Most of the low-cost 8K boards can be addressed only as a contiguous 8K block, and then only at the natural 8K boundaries. The Econoram II is unique in that, instead of being organized as a single 8K block, it's set up as two separate 4K blocks. Each 4K block can be addressed to any 4K boundary in memory. Selection of the address is done with two dip switches (one for each block) located at the top left of the board. This means you can change the address without pulling the board out in most instances. The dip switches are the new long-handle type, which are very easy to operate.

$\overline{PWR}$ or MWRITE selection: Another unique feature of this board is that a slide switch at the top is used to select between either $\overline{PWR}$ or MWRITE for use as the write strobe for the board. Most often you will use MWRITE, but MWRITE is generated by the front panel in most systems, and not all of the "front-panelless" systems generate MWRITE. Then you would want to switch it over to $\overline{PWR}$. I have yet to see a memory board with this feature.

Vectored interrupt option: Most memory boards are protectable, which means you can't write data into them

until they're unprotected. The Econoram II is no exception, however, it has one added extra. As an option, the board will generate a vectored interrupt if a write attempt is made to a protected block. Each 4K block on the board may be protected individually.

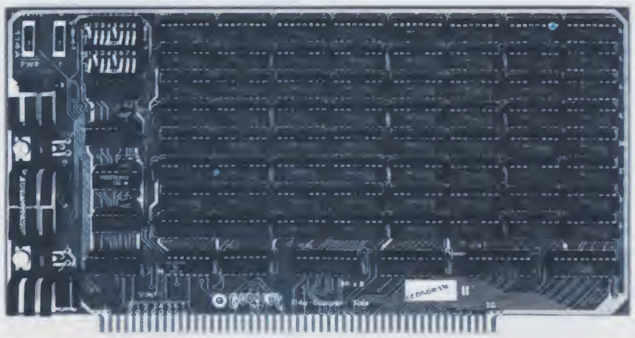
Wait states: It's rare to find a memory board these days with wait states on it. Most manufacturers boast, "No wait states required." Well that's true enough at 2 MHz. But I run my system at 4 MHz with a Z-80. A common misconception is that there's no advantage to running at 4 MHz if you don't have fast memory. Nothing could be further from the truth. Let me digress a bit: The processor is actually only accessing memory approximately 30 percent of the time. The rest of the time it's doing internal stuff. Consequently, if we're only waiting an average of 30 percent of the time, then 70 percent of the time we'll be zipping right along. The result is an effective 70 percent increase in speed, which is nothing to sneeze at!

The point of all this is that most of the memory boards don't allow you to insert wait states because they don't need them at 2 MHz. But the Econoram II allows insertion of a wait state. A slide switch located at the top left of the board selects 0 or 1 wait states.

Fast access time: While we're on the subject of speed, let's talk for a bit about chip access time vs. board access time. Access time is defined as the time it takes from the point the address data to the chip or board is stable, to the point at which the data you want to read or write is valid.

Take notice that the access time of the chip itself may not be the same as the board it's on. In other words, the board may introduce further delays increasing the total access time. Therefore, the board delay time plus the chip access time equals the board access time.

Bill Godbout guarantees the Econoram II to have a board access time of 450 ns. The chips themselves are 350 ns parts. But here's the amazing thing: In reality the chips are much faster. My board runs fine at 4 MHz with *no* wait states! Now Bill said that I could print that, but he *won't* guarantee it. But I know that my board is not just a fluke because I've heard reports of this fact from other people. By the way, it was in a fully loaded Imsai with a fan, a Cromemco 4 MHz CPU, running an exhaustive memory test for 12 hours with not one failure.



The Godbout Electronics Econoram II. Note the wait state, write strobe and address select switches in the upper left-hand corner. Photo courtesy Godbout Electronics, Oakland CA.

Although this is not a guaranteed spec, it's a little added extra bonus that makes the Econoram II a cut above the others. It's also a testament to the excellent design that's gone into this board.

Bypass Capacitors: An unwritten rule (well, maybe it's been written) of logic design is to take X amount of chips, mix in a handful of bypass capacitors, and shake well. Bypass capacitors keep noise and glitches from the power supply lines out of the chips. If you want maximum system reliability, you need adequate bypassing.

One new inexpensive 8K board I saw recently had not one bypass capacitor on it. I'm happy to say that the Econoram II has more than adequate bypassing.

Fully socketed: I don't want to get into the argument of whether or not to use sockets. I will admit that lousy sockets are worse than none at all, but good sockets are another story.

The Econoram II comes with a complete set of T.I. low profile, high quality sockets. This means you get a socket for every chip, not just the memory chips.

Of course, if you're anti-sockets you can leave them out. In any case, they are provided.

Assembly: Like most memory boards, the Econoram II goes together quickly. Mine took about two hours, but I'm rather experienced at putting these things together. Unfortunately, the assembly instructions are not of the Heathkit step-by-step nature. They read more like: "Mount resistors R1 and R2; then solder, and remove excess lead lengths." But since the board has a fully silk-screened legend on the component side, the minimal instructions are more than adequate.

The board has an excellent solder mask (that green goop) that makes it next to impossible to create solder bridges between adjacent traces. The board itself is a high-quality FR-4 epoxy type with gold-plated edge contacts. FR-4 is a flame retardant equivalent to G-10 epoxy.

Summary of Features

To sum up, the main features of the Econoram II are as follows:

1. Fully static.
2. Fully buffered.
3. Addressable in two separate 4K blocks by two long-handle dip switches.
4. Selection between $\overline{\text{PWR}}$ and MWRITE for board write strobe.
5. Write to protected memory can generate a vectored interrupt.
6. One wait state selectable for use at 4 MHz.
7. *Board* access time guaranteed at 450 ns.
8. Lots of bypass capacitors.
9. Low power dissipation — under 1.5 Amps.
10. Fully socketed.
11. Both sides solder masked.
12. Silk-screened component placement legend.
13. One year warranty.

I should mention that I am in no way affiliated with Godbout Electronics. I've never even bought so much as a chip from them before. Godbout Electronics in no way solicited this review. I bought the board (two, actually) for use in my own systems. To give you an idea of my taste in boards — most that I now have are either Processor Technology or Cromemco.

Econoram II is available from: Bill Godbout Electronics, Box 2355, Oakland Airport CA 94614 for \$150 assembled; in kit form, you can purchase four for \$475 (four is the limit), plus shipping (California residents, please add sales tax). Godbout's 24-hour phone order line is (415) 562-0636. They also accept Mastercharge and BankAmericard.

If you're not convinced by now that the Godbout Electronics Econoram II is one of the best memory buys on the market today, you really have to be one tough cookie — either that or you work for someone else who makes memory boards. ■

The TRS-80: how does it stack up?

The Radio Shack TRS-80 displaying part of a program listing. The computer's ac power supply can be seen behind the video monitor.

Ed Juge
2000 Thousand Oaks Dr.
Burleson TX 76028

Several years ago, my business purchased a Hewlett-Packard 9830. H-P called it a programmable calculator, but it was a microcomputer with H-P BASIC in ROM, and it was slightly larger than an electric typewriter. We plugged it in, turned it on, and it was an up-and-running computer. The price? Just about $14,000.

I knew when I bought an Altair 8800 a couple of years ago that it wouldn't be quite that easy. After I built and debugged it, there would still be a lot of accessories to buy — keyboard, output device, etc. — and after they were all operating I'd still have to make them work together, teach the 8800 BASIC, and spend some time working with it before I could actually run in BASIC. Not having a background in computer hardware or digital electronics, and finding people

who did few and far between, it took more time than expected. Through the efforts of several friends, and one in particular, the 8800 finally ran in BASIC about a month ago — some $1500 after my initial decision to buy.

All of this has nothing to do with the Radio Shack TRS-80, except to explain why I was so excited when I heard of its introduction, and why I ordered immediately and received one of the first 60 or so off the production line. You see, kits and assemble-your-own-system computers are beautiful for hardware buffs, but my one and only interest is programming.

The first surprise from Radio Shack was delivery within *six days* of the original expected delivery date. Astounding for a new item, whose target date was set before a production line even existed. The TRS-80 came well packed and consisted of the keyboard, video monitor, cassette recorder, ac power pack for the computer and two tapes — one blank and the other containing the prerecorded games Blackjack and Backgammon. Also included was a 30-page preliminary instruction book and a card saying that the 300-page owner's manual would be mailed to me in a few weeks.

Second surprise: everything plugged together easily; I connected power, threw the switches, typed in NEW, and it was a complete system, up and running in BASIC . . . *the first time*! Just like my $14,000 H-P had been, but for $599.95!

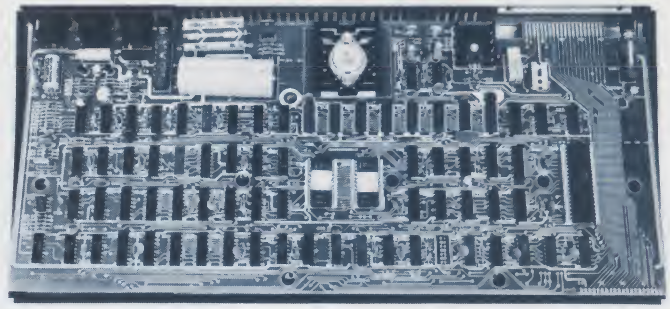
The TRS-80's CPU, 4K ROM, 4K RAM, video and cassette interfaces — everything — are housed in the keyboard enclosure. No plug-in boards. No motherboard. No heat problem. No twenty-pound power supply. The TRS-80's external power supply is hardly larger than a large capacitor we had to add to the Altair to make it handle an extra memory board. The housing is beautifully styled in gray and black. The video monitor is in a matching cabinet, making it unusually attractive in any setting.

Should you wish to expand your TRS-80 system, there is a small door in the rear of the keyboard enclosure which covers a 40-pin connector for later addition of peripheral equipment. A RESET button is also under this door. It permits you to "unlock" any condition (such as an unending loop) that can't be unlocked from the keyboard BREAK key. RESET unlocks without disturbing your memory.

The keys look as if they were taken from an IBM Selectric typewriter, and the touch is excellent. Either a couple of them aren't as positive as they should be, or I get a bit light-fingered on them, as I occasionally miss them when typing fast.

Radio Shack was thoughtful enough to provide a nice cassette recorder with its own built-in ac power supply. . . no batteries to buy unless you want to. The TRS-80 starts and stops the recorder on either the CLOAD or CSAVE commands. Should you want to rewind or run fast forward, you must unplug the "remote" cable to the TRS-80. An override momentary-contact button on the recorder would have been more convenient and saved wear and tear on the tiny jacks and plugs.

The video monitor is a 12-inch unit with very good display, and is easy to read under any room-lighting conditions. If you're thinking that a $199.95 video monitor is quite expensive just for a matching cabinet . . . well, maybe so — until you rush home from work anxious to do some computing, only to find the portable TV you've modified is being used by a roomful of kids to watch the Gong Show. You won't have to worry with the TRS-80's monitor; it has no tuner, so won't tune in TV. Your output device will stay an output device.



Bottom view of TRS-80 shows board containing CPU, 4K ROM, 4K RAM, video logic and cassette and video interface hardware. Plugs for cassette recorder, video monitor and ac power pack are at upper left with power on/off switch. Reset switch is at upper right next to 40-pin connector for peripheral equipment.

Operation

Until now, I have never seen a hobby computer you could carry home from the store, plug in and use in anything other than machine language (numeric codes). The manufacturers apparently have been so deeply into computers themselves that they failed to realize how many people out there in the real world have no experience or training with computers, aren't interested in constructing a unit or having to program with lights, switches or even numeric codes . . . but who would jump at the chance to own a computer they could talk to in near "plain English" languages such as BASIC.

There certainly is a market composed of hardware buffs with training and experience in the field, but how many more of us must there be in the nontechnical part of the market? Well, obviously Radio Shack is going to find out, because we're the ones they've built the TRS-80 for . . . along with small businesses and the educational market. In shooting for our market, however, there is one essential ingredient other than an up-and-running machine: training the complete novice to program a computer.

Radio Shack's preliminary manual is one giant step in the right direction. If their 300-page book is an expansion of the small one, they should have the problem solved. The preliminary manual not only gives you complete initial hookup instructions, but it gives you a thorough course in the basics of programming. The manual starts with simple arithmetic calculations and progresses through each of the TRS-80's functions, commands and statements. Every step of the way, the manual spells everything out in simple, nontechnical language, and it gives you sample programs to try as you read, so you will see for yourself how the TRS-80 operates, given a particular command. My fifteen-year-old went through the entire procedure with no trouble or questions arising that he couldn't answer himself. The preliminary manual indicates that the 300-page book will contain a detailed, complete programming course in Radio Shack Level I BASIC.

If you're interested in seeing the TRS-80 run immediately, without having to go through this short course in programming, their game tape will have you playing Blackjack or Backgammon in a couple of minutes.

In operation, the TRS-80 appears to be reliable and trouble-free. Mine has been on for a period of 17 hours straight at least once, and I've probably logged a minimum of 40 hours so far. During that time, operation has been flawless. To offer a comparison — I've seen the time when BASIC had to be

loaded into the Altair (via paper tape) ten times before it was done without some type of error showing up in the operation. The H-P 9830 could be expected to show some kind of problem in that many hours (an H-P salesman once said that they anticipated one error in something like every 15 million calculations. These weren't errors in calculations, but in tape loading or storing, memory dumps, etc.).

So far, the TRS-80's operation has been outstanding by these standards. I've found no problems from power-line surges (as when the air conditioner comes on), dust, carpet fuzz or anything of this nature, to which some units are hypersensitive.

Radio Shack might look into some additional rf filtering, however, as my TRS-80 does slightly affect the color on a TV operating on rabbit ears in the next room. I suspect (because the level of interference is so low) that a good outdoor TV antenna would cure the problem.

The advertised 4K of RAM, my computer tells me, is exactly 3583 bytes. Should you anticipate needing more memory, additional plug-in memory is available for your TRS-80 at your local store.

Most hobby computers do nothing when first turned on except operate in binary codes. Radio Shack's 4K BASIC in ROM makes the TRS-80 an intelligent and conversational machine as soon as power is applied. If you've experienced the other type of operation, you're in for a pleasant surprise!

Radio Shack Level I BASIC

Radio Shack's 4K Level I BASIC appears to pack an awful lot into 4K. It includes most standard BASIC commands, video handling and graphics. It does not, however, include trig functions. Commands available are NEW, LIST, RUN, CONTinue, REMark, LET(optional), GOSUB-RETURN, FOR-NEXT-STEP, GOTO, IF-THEN, INPUT, ON ... GOTO, ON ... GOSUB, PRINT, CSAVE, CLOAD, DATA, READ, RESTORE, STOP and END. Functions are: ABS, RND, INT, TAB, +, -, X, ÷, >, <, <=, >=, and =. The command MEM, which tells you the amount of RAM that remains unused at any time, is also provided.

Graphics are also available, and their special commands are: CLS (clear screen), SET (X,Y) (turns on point X, Y), RESET (X,Y) (turns off point X, Y) and POINT (X,Y) (used with IF . . . THEN, and determines if X, Y is set).

The non-graphics display is made up of 16 64-character lines, and this display is mapped with 1024 positions. You can instruct the TRS-80 to PRINT AT (position #) to place your display anywhere on the screen you wish without the need for printing a bunch of blank spaces or lines to get there. The graphics display consists of 128 horizontal and 48 vertical positions, and, of course, graphics and text can be employed together.

The TRS-80 has only three error codes:

WHAT — the computer does not understand your instruction.

HOW — the instruction is understood but cannot be executed (such as GOTO a nonexistent line number).

SORRY — you've used all of the available memory.

When you've programmed in an error and RUN the program, the computer will reach the error, stop, and display the error code followed by the line in which the error occurs. A question mark appears in the faulty line at the exact position of the error, making debugging a program simple and straightforward.

Only two string variables, A$ and B$, are available in Level I, and each is limited to 16 characters. The single array A(n) can be used, and is limited only by the memory capability of your TRS-80. Variables A through Z are permitted. Integer limit is 32,767, which is also the limit of line numbers. Large or very small numbers are handled in scientific notation.

Most program statements and commands can be abbreviated, and statements can be stacked on a single line. Thus a program that reads:

```
10 FOR X = 1 TO 100
20 PRINT X, 2X, X/2
30 NEXT X
40 END
```

could be written as shown in Example 1 (END has no abbreviation). The "long" version requires 51 bytes of memory, whereas the abbreviated version takes only 33 bytes — a saving of 18 bytes or 35 percent. This can be significant in long programs.

```
10 F.X=1TO100:P.X,2X,X/2:N.X:END
```

Example 1.

One of the more interesting features of the TRS-80 is its excellent random number generator. A random number is generated by the command RND(X), with a number being returned between 0 and X. A random number between 0 and 1 will be produced by the command RND(0). The numbers returned are truly random, and no repeating sequences have been observed, even from initial RUN of an RND command. Combining RND commands with the graphics capability of the TRS-80 can provide hours of fun. A few samples of such programs are included in the preliminary manual.

A cursor (_) is always present on the monitor, indicating where the next display will appear. During programming, it can be backspaced with a ← key to correct errors within that line. Correcting a line after entry requires retyping the entire line. Eliminating a line is done by typing the line number only and ENTER. The TRS-80 puts all lines in proper sequence, regardless of the order in which they are entered. The LIST command puts the program (first 16 lines) on the screen, and a ↑

TRS-80 during CLOAD. Note two asterisks in upper left corner of screen. The right one blinks on and off during cassette loading if program is loading properly.


XITAN

key scrolls it up for viewing additional lines.

There are two additional keys related to the two mentioned above. These are → and ↓, both of which are inoperative. The preliminary manual does not mention them, although it would have saved Radio Shack a lot of phone calls from customers who are afraid they have a defective unit. My call to them confirmed my suspicions that these keys are for use in their 12K BASIC, which should be available before the end of 1977. This was the only note that I felt should have been included in the preliminary manual but wasn't. The 12K BASIC will be available in plug-in internal ROM and is expected to cost around $130, installed by your local Radio Shack. A complete listing of Level I BASIC is contained in the back of the manual. Also included is an explanation of each command and its abbreviation.

For those of you who worry about your computer "going to sleep," Radio Shack has included in the upper corner of the display a couple of asterisks that appear during CLOAD. If the cassette is loading properly, one of them will blink on and off. When you're storing a program on tape (CSAVE), recording volume level is automatic. When you're loading a program from tape into the TRS-80 (CLOAD), the recorder's volume control operates. Its setting should be (according to the manual) between 7 and 8. I found this critical on the prerecorded game tapes, as their level is somewhat low. For programs I've recorded on tape, playback is perfect at any volume setting above 3.

Operating Impressions

Using any micro for a while is bound to leave you with predetermined ideas about what a good machine should and shouldn't do . . . features and commands it should or shouldn't have. These ideas are based largely on an individual's use of a computer. Trying to allow for my preconceived ideas (and hoping 12K BASIC will include some of them), I must say that the TRS-80 has to run at the top of the list for anyone wanting an up-and-running, no-hassle, nothing-to-build, nothing-else-to-buy personal computer — without even considering the low price.

It runs in BASIC as it comes from the carton. Reliability couldn't be better and its size makes it useful anywhere. Styling would be difficult to improve on. Assuming Radio Shack plans to arrange for at least limited local (plug-in) service, finding a local Radio Shack store should be easy in most places. Where else can you get an up-and-running machine with a manual that can teach even the complete newcomer how to program?

There's always the question of speed in a computer. To most of us, the only reason to speed up a machine capable of about 5400 simple calculations per minute (that's around 90 per second) would be to boost our egos. I can think of few practical applications of a hobby computer that would be disadvantaged by that speed. That is approximately the speed of the TRS-80.

Kilobaud published the results of some benchmark timing tests in issue No.10, comparing the speeds of 33 microcomputers. In case you're interested, the TRS-80 stacks up as follows:

Test No.	Seconds
1	2.5
2	18.6
3	24.4
4	37.0
5	46.0
6*	
7	110.1

* Test No. 6 included a DIM statement not available in Level I BASIC.

This places the TRS-80 in the slower third of the machines tested, while other

Keyboard with its circuit board, and partial rear view of TRS-80's main board.

computers using the same Z-80 CPU chip scored in the top third. The difference could be a combination of the clock speed Radio Shack uses in the TRS-80 and/or their Level I BASIC. It's possible that their 12K BASIC will show faster times. It's reassuring to know, however, that *you* are smart enough to give your computer a problem that it has to think about for a few seconds sometimes! Maybe man still has the upper hand over machine.

Many people ask, "So you've got a computer . . . whaddya do with it?" This could be an embarrassing question. Everyone's answer will be different. Unless you're an engineer or businessman who can use one to solve equations or keep inventory records, the question of practicality of use is questionable. Don't let this discourage you . . . how really, truly practical is your television set most of the time?

My enjoyment in using a computer comes from programming: solving problems . . . whether or not I have any use for solutions. It's fantastic exercise for your mind, which I enjoy. Just writing a program that runs is only half the fun for me. After my program runs, I like to see if I can make it run faster by altering or improving the programming — or make it do the same thing with fewer steps. I doubt that many people will ever find themselves writing the best, fastest possible routine on the first try.

In The BASIC Forum (*Kilobaud* No. 9), a problem is presented: Write a program asking you for a number (n). The program then locates and prints all prime numbers between 1 and n. My first try required over 30 seconds to search from 1 to 100. My first revision cut the time in half.

The best I could do was about 11.4 seconds without the printout containing an error because of my programming. I don't want to admit the time spent on it, but it occupied several fun evenings. I sincerely hope that will become a monthly undertaking in the forum. The authors solicit readers' programs, and will publish the best ones so you can see how your programming skills are shaping up.

Some Applications

Being a ham radio operator, I have used my computer to calculate all possible combinations of frequencies (channels, if you prefer) in the 2-meter FM band that will combine to produce an unwanted signal on any given (other) channel. The program is simple and short, but since

you can have a multiple of one channel combining with a different multiple of another, if you check all possible combinations up to the fifth multiple of each, that simple program may take half an hour to run, and involve 9,000,000+ calculations. Try that with your pocket calculator! The results could be useful under certain circumstances, and writing the program was interesting.

Practical applications? Well, if you must feel your machine is practical, and if you don't think making it do useful work will ruin its standing as a hobby computer, you can have it keep your family financial records; store phone numbers, addresses, or both; file recipes; remind you of birthdays and anniversaries; almost anything. Just use your imagination. Radio Shack may be a step ahead of you on some of these uses — they have cassette program packages available for small-business payroll, teaching the kids math, kitchen uses (menus, conversion tables, directory and message center) and personal finance. These vary from a single cassette to a seven-cassette portfolio, and prices range from $4.95 to $19.95. More are planned.

Just because the TRS-80 is a one-circuit-board unit without plug-in accessory boards, don't think you're stuck with what you've bought. This year Radio Shack plans to have a couple of accessory printers, a disk memory, provisions for a second cassette recorder and the 12K BASIC already discussed. Right now you can get additional internal plug-in RAM options of an additional 4K or 12K. Planned expansion also includes serial I/O, parallel I/O, an expansion unit for additional PC boards and a modem. They're considering digital cassettes, enhanced graphics, a music synthesizer, external device controller, color graphics, a 14-key numeric pad, and who knows what else.

Unquestionably, Radio Shack's TRS-80 has given a giant boost to their image, and provided — at last — a personal computer for the masses: trained or untrained. To those among you who say such a package unit, ready to run, requiring no hardware knowledge, limiting the haywiring you can do, isn't in the true computer-hobby spirit, I have one thing to say: Wanna buy an up-and-running 12K Altair at a bargain price?

To the rest of you: Anybody for a TRS-80 users' group? ■

A Tale of Four BASICs

which one for *your* 6800?

Rich Didday
1218 Broadway
Santa Cruz CA 95062

Rich has provided us with a rather gory look at what you'll have to go through if you're foolish enough (as he was) to try implementing foreign BASIC interpreters on your machine. But, the primary intent of his article is to provide a review and comparison of four popular BASICs for 6800-based machines . . . and he does an excellent job. I'd like very much to see other articles along this line comparing 8080-based BASICs, assemblers, text editors and others. — John.

I have four different versions of BASIC up and running on my 6800-based system, and I have the feeling that I might have a few more before I'm through. I'm not collecting BASIC interpreters as if they were old coins or something — it's just that I want to find a version that really suits me. What am I looking for? I want a version that: 1. will run any program written in ANSI Minimal BASIC (see Box 1); 2. is convenient to use; 3. is reasonably fast; 4. is reasonably inexpensive.

Besides that, I've just been curious about what sort of software products is becoming available.

I thought I'd share my experiences with you, so that if you're thinking of buying a version of BASIC, you might find some of your questions answered here — before you send in your hard-earned cash.

There's a wide range of differences among the four versions, and, depending on your priorities, almost any one might be best for you. I'll cover every aspect I can think of, from cost to speed to documentation to numerical accuracy to range of statements provided, etc.

The Four BASICs

Table 1 shows the four BASICs, with the latest prices I could find. I've listed them, and I'll discuss them, in historical order of their appearance on my system.

As you may have guessed by looking at the price, I didn't purchase Sphere's provisional nonextended BASIC — it came with my system. It was the first version I had up and running, the one I've used the least, and the main force behind my search for other versions.

Next I ordered a copy of Pittman Tiny BASIC; it showed up immediately (less than two weeks). Since it's available only on paper tape, I had to get my friend Nick

Version.	Manufacturer.	Cost for paper tape.	Cost for cassette (KC Std.).
Sphere provisional nonextended BASIC	Sphere Corp. 791 South 500 West Bountiful UT 84010	$325[a]	$300[a]
Pittman Tiny BASIC	Itty Bitty Computers P.O. Box 23189 San Jose CA 95153	$5	
SWTPC 8K BASIC Version 1.0	Southwest Technical Products Corporation 219 W. Rhapsody San Antonio TX 78216	$20	$9.95
TSC Micro BASIC Plus Version 2.1	Technical Systems Consultants Box 2574 W. Lafayette IN 47906	$6 + $15.95[b]	$6.95 + $15.95[b]

[a]This is the price listed for "BASIC Version 1" in the June 1976 *Global News* (Sphere's newsletter). I believe that this is the "provisional nonextended BASIC" that comes with the Sphere 330.

[b]The documentation for TSC Micro BASIC Plus is priced and sold separately.

Table 1. The four BASICs.

BASIC is a higher-level, interactive computer language that was originally developed at Dartmouth College in the middle 1960s. The design goal was to create a reasonably powerful language that would be particularly easy for beginners to learn and use.

It is easy to learn, and it is also easy to implement. That is, compared to other higher-level languages, it is relatively simple to write a program that accepts statements in BASIC and carries them out. This program (which is a machine-language program tailored to the system it is to run on) is called an *interpreter*. An interpreter differs from a *compiler,* which is a program that accepts statements in some higher-level language and translates them into a machine-language program that can be run later to carry out the instructions of the original higher-level program. An interpreter, on the other hand, translates each statement of the higher-level program each time it comes to it, and carries it out immediately.

If you buy one of the four versions of BASIC discussed in this article, you get a copy of a machine-language program (i.e., the interpreter) plus documentation that describes what you got, how to tailor it to your system and how to use it. More abstractly, you also get the capability to enter, alter and run a broad variety of programs in some variant of the higher-level language BASIC.

There are, then, two different sorts of questions you need to answer in order to decide which version is best for you.

1. Questions about the interpreter, considered as a machine language program in and of itself. For example: Will the interpreter run properly on your system. Will it be hard to adapt to idiosyncrasies of your system (will it be hard to install)? How much memory space will it need? What's the cost?

2. Questions about the language the interpreter accepts. For example: What statements are allowed? What data types are allowed? How fast will a given program be carried out? How convenient is it to enter, alter and run programs using this interpreter? How useful are the error messages?

The answers to these questions are not all simple. Here's just one example. You might think that the more statement types a version of BASIC lets you use, the better; or that the more statements stolen from FORTRAN or ALGOL or whatever your *real* favorite language is, the better. In fact, every vendor sticks fancy little things in his version to try to attract and hold you. But what if one of your goals is to be able to trade programs with other people? What if you don't want to do much programming yourself, but want to buy programs to run on your system? Then anything but the fairly well-agreed-on, "standard" statements in your BASIC will be useless.

This factor itself would be easier to judge if there were an agreed-on standard version. The American National Standards Institute (ANSI) has been working on a set of standards for what they call "Minimal BASIC" for some time now. (A draft of the proposed American National Standard Programming Language Minimal BASIC, report BSR X3.60, X3J2, January 1976, is available from: CBEMA/Secretary X3, 1828 L St. NW, Washington DC 20036.) Their plan is to define a minimal subset, and then define a series of "enhancements" covering such things as string manipulation, matrix operations, file manipulation, etc. When they finally agree on the standards, we'll be in better shape — you'll be able to tell exactly what's included in a version if it says "ANSI Minimal BASIC with string enhancement." Right now, you're on your own! Good luck.

Box 1.

to make a copy on cassette tape. I got it flying with relatively little effort, and have used it to write a fairly lengthy game program. It's a carefully done piece of software, and I'd use it more if I didn't care about being able to write programs that conform to the proposed ANSI standards.

The third version was Southwest Tech's 8K BASIC. I knew when I got it that it was made specifically for the SWTPC 6800 machine, that it didn't come with a source version of the interpreter, and that it would certainly be a lot of work (or maybe even impossible) to convert it to my system; but I felt like a challenge, or else I was desperate (I don't remember which). As it turns out, it's the version that I use most.

The fourth is TSC's 6800 Micro BASIC Plus, which, as the name implies, lies between Tiny BASIC and a full version like SWTPC's 8K in size and capabilities. It arrived instantaneously (less than one week).

Documentation

There's quite a range here, all the way from virtually nothing up to a 68-page booklet describing the product inside and out (see Table 2).

The documentation provided with Tiny BASIC, SWTPC 8K, and Micro BASIC Plus is generally very good. Each gives thorough descriptions of each statement type. They're all good about giving little examples of the use of each construct, they all list error message meanings, and they all give lists of the memory addresses of important stored variables (start of user program, entry points to the interpreter, end of memory pointer, etc.).

I've programmed in BASIC for quite a while, so it's hard for me to be certain, but I *think* someone who had never programmed before might be able to pick up enough from the Tiny BASIC documentation to be able to write programs. The others assume that you know something about programming, and I guess Sphere assumes you're clairvoyant.

Range of Statements Provided

Table 3 shows the statements that are provided by each of the four versions. You can probably figure out what most of them do. If not, see Table 1 *in* Stephen Pereira's "Now It's Imsai BASIC!" *Kilobaud* No. 5, May, 1977.

Though all four BASICs have quite a few statement types in common, there are some slight differences. All *provide a limited* text-editing capability, which lets you enter (numbered) lines, list

Version	Number of pages of documentation	Contents
Sphere provisional nonextended BASIC	3	A list of legal keywords and operators plus two pages telling how to load the tape.
Pittman Tiny BASIC	26	Clear descriptions of statement types and meanings, three pages of sample programs, instructions for installation on seven different systems.
SWTPC 8K BASIC Version 1.0	26	Clear descriptions with small examples of all statements.
TSC 6800 Micro BASIC Plus, Version 2.1	68	Clear descriptions of all statement types with examples, lots of useful little side notes, plus complete listing of assembly language source text (well commented).

Table 2. Documentation.

them to see what you've got, and wipe out or alter existing lines. Since many, if not most, home users will be employing TVs or other video displays, it's important to be able to list only a specified number of lines. The typical form for doing this is: LIST 30,120 — which lists all lines from number 30 up to and including number 120. All except Sphere allow this form. As nearly as I can tell, there is no way in Sphere's version to have a look at the first part of a long program. Forms like LIST 30 are allowed, but list all statements from line 30 through the end of the program. Even though Sphere does everything slowly, it doesn't list slowly enough for you to read the program on the fly.

Every version provides a way to wipe out the program you've been working on and start a new one. In Sphere and TSC, you type SCRATCH. In Tiny BASIC, you type CLEAR; and in SWTPC 8K you type NEW.

Two versions have a command that returns control to the host computer's monitor. PATCH does this in SWTPC 8K; MONITOR does it in Micro BASIC Plus. Tiny BASIC doesn't have a separate command for it, but it's not hard to put together two calls to the USR function (which carries out machine language routines for you) to accomplish it (and Pittman's documentation makes it perfectly clear).

In Sphere, the only way to get back to the monitor (or even to just interrupt an executing program) is to (shudder) reset the machine. On my system, that means you're flirting with disaster, since hitting the reset switch stops the dynamic memory refresh. After a while, you get pretty good at getting your finger off the reset button quickly, but still . . . .

Both TSC and Pittman tell you carefully and completely what to do if you don't like their choices of breakpoint, back space and prompt characters. Specifically, they tell you where those characters are stored, so you can insert your own choices if you wish. SWTPC doesn't give you that information, and you're stuck with what they give you. Sphere, believe it or not, doesn't even have back-space capability. If you make a mistake in typing, there's no way to go back; you just have to hit return and wait for the thing to give you an error message!

SWTPC 8K comes closest to being able to run any program that conforms to the (soon to be adopted) ANSI standards for Minimal BASIC. SWTPC's documentation points this out and notes one disagreement with the standards — all arrays start at location 1 in SWTPC 8K. (In the proposed standards, it is the programmer's option whether arrays start at 0 or 1.) I haven't made a detailed check of SWTPC 8K against the standards, but since it is claimed that the array starting point is the only conflict with the standards, I'll mention a few other conflicts I chanced across.

The first *might* simply be a bug I introduced when I installed the interpreter on my system. At any rate, according to the proposed standards,

```
20 PRINT "LINE 1",
30 PRINT "AGAIN"
40 END
```

Sphere provisional nonextended BASIC

Commands	Statements	Functions	
RUN*	DATA	ABS	
LIST*	DIM	ATAN	
SCRATCH*	DEF	COS	
	END	EXP	
	FOR NEXT STEP	INT	
	GOSUB	LOG	
	GOTO	RND	
	IF THEN	SGN	
	INPUT	SIN	
	LET	SQR	
	MAT	TAN	
	PRINT		
	READ	CON	
	REM	IDN	
	RESTORE	INV	matrix
	RETURN	TRN	operations
	STOP	ZER	

TSC Micro BASIC Plus

Commands	Statements	Functions
RUN*	DATA	ABS
LIST*	DIM	RND
SCRATCH*	END	SGN
MONITOR*	EXTERNAL*	SPC
	FOR NEXT STEP	TAB
	GOSUB*	
	GOTO*	
	IF THEN	
	INPUT	
	LET*	
	ON GOSUB	
	ON GOTO	
	PRINT*	
	READ	
	REM	
	RESTORE*	
	RETURN	

SWTPC 8K BASIC

Commands	Statements	Functions	
RUN*	DATA	ABS	
LIST*	DEF	COS	
NEW*	DIM	EXP	
PATCH*	END	INT	
SAVE*	FOR NEXT STEP	LOG	
LOAD*	GOSUB*	PEEK	
APPEND*	GOTO*	POS	
LINE*	IF THEN*	RND	
DIGIT*	INPUT	SGN	
	LET*	SIN	
	ON GOSUB*	SQR	
	ON GOTO*	TAB	
	POKE*	TAN	
	PORT*	USER	
	READ*		
	REM	ASC	
	RESTORE*	CHR$	
	RETURN	LEFT$	
	STOP*	LEN	string
		MID$	operations
		RIGHT$	
		STR$	
		VAL	

Pittman Tiny BASIC

Commands	Statements	Functions
RUN*	END*	RND
LIST*	GOSUB*	USR
CLEAR*	GOTO*	
	IF THEN*	
	INPUT*	
	LET*	
	PRINT*	
	REM*	
	RETURN*	

Table 3. The commands, statement types and functions provided by the four BASICs. Those which can be used in the direct mode, i.e., without being part of a program, are marked by an asterisk.

```
1000 PRINT "I ";
1010 FOR S=1 TO -1
1020   PRINT "DON'T ";
1030 NEXT S
1040 PRINT "CONFORM TO THE ANSI STANDARDS"
1050 PRINT "CONCERNING FOR LOOPS."
1060 END
```

Example 1.

```
I DON'T CONFORM TO THE ANSI STANDARDS
CONCERNING FOR LOOPS.
```

Example 2.

should yield a single output line when run, like this:

```
LINE 1            AGAIN
```

but (on my system at least) the program prints on two lines:

```
LINE 1
AGAIN
```

Another conflict with the standards concerns FOR loops. Running the program in Example 1 in SWTPC 8K produces the output in Example 2.

If the standard had been followed, the "DON'T" would not have been printed. This is because the initial value of S is already greater than the upper limit (-1).

Finally, the proposed standards decree that the random number generator function (RND) not take an argument, and that a command RANDOMIZE, which initializes RND, be included. In SWTPC 8K, RND must be given an argument, as in

```
LET X=RND(0.0)
```

and RANDOMIZE is not implemented.

Overall, though, SWTPC 8K is the only one of the four that comes close to the proposed standards.

Sphere comes next closest to being able to run any legal ANSI Minimal BASIC program; neither Tiny BASIC nor Micro BASIC Plus makes any real pretense of trying to conform to the proposed standards. Their design strategy calls for providing as much power for the user as possible within a severely limited memory space, and that leads both of them away from standard forms of BASIC. If your system has enough memory to run something like SWTPC 8K, you can decide how important it is for you to be able to run standard BASIC progams. If you have only 4 or 5K of memory, having a full Minimal BASIC is a frill you can't afford.

Sphere's version does have some matrix operations (see Table 3). For example, these statements

```
10 DIM A(20,20)
20 MAT A= IDN
```

store the identity matrix (1s on the diagonal, 0s everywhere else) in the array A. Interestingly enough, these statements

```
10 DIM A(20,20), B(20,20)
20 MAT A= INV(B)
```

do the same thing, no matter what B is. The documentation says that INV stands for inverse, but it has the same effect as IDN. Oh, well.

SWTPC 8K includes a number of string operations, and allows arrays of strings. I personally find these features extremely useful. SWTPC also incorporates the commands SAVE, LOAD, APPEND, which (respectively) store an active program on tape, load a program from tape and append more lines to an existing program in the machine. None of the versions includes an explicit way to store and retrieve data on tape — SWTPC makes it reasonably easy, though.

Memory Requirements

There's a big variety in the memory requirements of the four BASICs. Obviously, versions (like SWTPC) that include many different legal commands are going to require more memory than those with a more limited repertoire (like Tiny BASIC). If your main constraint is limited RAM, Table 4 may be enough for you to choose which of the four is right for you.

In terms of memory (as well as other requirements), Sphere is in a category all its own. Apparently, it's actually

Version.	Approx. memory space required for interpreter alone.	Suggested minimum memory.	Memory required for medium-sized programs.
Sphere provisional nonextended BASIC.	15.2K	20K	20K
Pittman Tiny BASIC.	2.5K	3K	4K
SWTPC 8K BASIC Version 1.0.	7.1K	8K	10K
TSC 6800 Micro BASIC Plus, Version 2.1.	3.3K	4K	5K

Table 4. Memory requirements of the four BASICs.

Version.	Internal representation.	Number of digits displayed.	Decimal Digits of accuracy.	Largest number.	Smallest number greater than 0.
Sphere	floating point, binary, 4 bytes per value	6	4 or less	5.793E76	4.0E-78[1]
Pittman	fixed point, binary, 2 bytes per value	5	4+	+32767	1
SWTPC	floating point, BCD, 6 bytes per value	9	9[2] or 6[2]	9.99999999E+99	1.E-99
TSC	fixed point, BCD, 3 bytes per value	5	5	+99999	1

[1]Sphere BASIC has a number system all its own — each different scheme I hit on to determine the smallest number greater than zero produces a different result, some with exponents of -78, others with exponents of -77.

[2]Normal arithmetic (+, -, *, /) yields results accurate to 9 decimal digits. The built-in functions (like SIN, COS, EXP, etc.) give results accurate to 6 decimal digits.

Table 5. Number representations in the four BASICs.

a batch (as opposed to interactive) version lifted (with a few alterations made and bugs added) from another machine. Instead of translating that version into 6800 machine language, Sphere wrote an emulator for the original machine. So, when you run Sphere's version, you're actually simulating running BASIC on another machine. While this *is* a clever way to bring up a version of BASIC on the 6800 rather *rapidly*, it requires so much memory, runs so slowly (as we'll soon see) and is so inconvenient to use that it is acceptable only as a stopgap measure. Since other versions are available, it seems that Sphere BASIC has little to recommend it.

Arithmetic

As you can see in Table 5, two of the versions store numbers in binary coded decimal (BCD), two use the more normal binary representation. The two larger versions offer floating-point numbers (numbers with exponents and fractional parts); the two smaller ones don't.

Both SWTPC and Sphere provide a range of arithmetic functions (SIN = trigonometric sine, COS = cosine, LOG = natural logarithm, SQR = square root, etc.). The SWTPC arithmetic functions are accurate to six decimal digits (operations like adding, subtracting, multiplying and dividing are accurate to nine decimal digits). SWTPC arithmetic functions are inordinately slow (more on this later). Most of Sphere's arithmetic operations and functions are accurate to four decimal digits (although they are displayed to six digits), but some of them are unbelievably inaccurate. For example, Sphere's LOG function has no digits of accuracy for arguments around 1.0 (see benchmark program functions in Table 6). And (see if you believe this one), 1/(-1) evaluates to .25!

All except Micro BASIC Plus implement the normal operator precedence rules. In TSC's version, arithmetic operations are performed left to right, unless parentheses are used to force another order. Thus, PRINT 1 + 2*3 prints the value 9 in TSC BASIC. It prints 7 (as you'd expect) on the other three versions.

Speed

There are quite a few factors that affect how fast a given program will run on different interpreters. For instance, you'd expect that, in general, BASICs that implement floating-point arithmetic with a large number of significant digits would run slower than versions restricting themselves to small integers. On the other hand, you'd expect that versions made to fit into a tiny amount of memory would be a bit slower than those that could afford to do more elaborate processing on the program before it's run. You'd expect BASICs that provide a large number of different statements to run a little slower, because it probably will take longer to decipher any given statement. And, of course, you'd expect a version that was run by simulating another machine would run many times slower than one written specifically for the 6800.

I tested the relative speeds by running nine benchmark programs. The first seven are those used by Tom Rugg and Phil Feldman in their recent article (see "BASIC Timing Comparisons," *Kilobaud* No. 6, June 1977). Table 6 shows the results of all nine benchmark programs.

Since my system runs with a slow clock, I've normalized all the resulting times. If your 6800-based system runs at full-rated clock speed (1 MHz), you should observe the times shown in Table 6.

There aren't really any big surprises in the times taken for the first seven benchmark programs. For each of them, Sphere is more than ten times slower than the slowest of the other three. Micro BASIC Plus seems to be a little faster than you might expect. One interesting point is that the standard BASIC assignment statement using the keyword LET, as in 40 LET A=0, runs faster than the nonstandard ones used by Rugg and Feldman, 40 A=0, at least in Tiny BASIC. None of the other versions would accept the nonstandard assignment statement. TSC's documentation says that the nonstandard form is OK, but on my particular system, using it yields obscure error messages.

Since Rugg and Feldman didn't run any benchmarks that tested numeric functions or string manipulations, I made up two additional benchmark programs. They're shown as Program 1 and Program 2.

The function benchmark (Program 1) measures the maximum absolute error encountered in the operations it goes through. Since

Time (in seconds) to run benchmark programs.[a]

	1	2	3	4	5	6	7	functions	strings
Sphere	166[b]	493	1325[c]	1207½[c]	1290[c]	4250[c]	6003[d]	8059[c]	--
Pittman	--[e]	41[f]/37[g]	61	62	83	284[h]	--	--	--
SWTPC	15[b]	25	96	105	109	173½	204	6631[c]	47
TSC	9[b]	19½	48	51	61	109	223½	--	--

[a]The first seven benchmark programs are those used by Tom Rugg and Phil Feldman in their article in *Kilobaud* No. 6, pages 66-70. Times shown are normalized to show the time required if the 6800 was run at its rated clock speed (1 MHz). To get the actual times observed on my system, multiply by 1.45 (e.g., TSC 6800 Micro BASIC Plus actually took 9x1.45 = 13.5 seconds to run benchmark program 1 on my system). Times are believed to be accurate within ½ second.

[b]This version of BASIC will not accept the nonstandard assignment statement (e.g., 400 K=0) used by Rugg and Feldman; so LET was inserted as necessary (as in 400 K=0).

[c]Program was actually run for 100 interations; elapsed time was multiplied by 10.

[d]Program was actually run for 10 iterations; elapsed time was multiplied by 100.

[e]Pittman Tiny BASIC does not implement FOR-NEXT loops.

[f]Using nonstandard assignment statements (i.e., without LET).

[g]Using standard assignment statements (i.e., with "LET").

[h]Pittman Tiny BASIC has no arrays, so the DIM statement in Rugg and Feldman's benchmark program number 7 was replaced by a REM statement. Also, the FOR-NEXT loop in their program was replaced by the equivalent counting loop.

Table 6. Speed.

take a close look at
PERIPHERAL VISION

Peripheral Vision is a young, fast-moving company that's dedicated to selling reasonably priced peripherals for various manufacturers' CPU's.

So now, when you build your microcomputer system, you'll know where to look for all the peripherals that will make your system do what it's supposed to do.

Peripheral Vision may be young, but we have some old-fashioned ideas about how to run our business.

We know there are serious incompatibilities among the various manufacturers' peripherals and CPU's. We want to get them together. And we want to bring significant new products to market--products consisting of everything from adaptation instructions/kits for hardware and software to major new designs.

Most important to our customers, Peripheral Vision is committed to helping you get along with your computer. We'll do all we can to make it easy.

Our first product is a real reflection of this philosophy. It's a full-size floppy disk for the Altair-Imsai plug-in compatible S-100 BUS. And it's available for as low as $750.00.

Our floppy disk has many exciting features:

- 1 interface card supports 4 or more drives
- Stores over 300,000 bytes per floppy
- Bootstrap EPROM included--no more toggling or paper tape
- Completely S-100 plug-in compatible
- Drive is from Innovex (the originator of the floppy concept)--assembled and tested
- Disk operating system with file management system included on floppy
- Cabinet and power supply optional

Also in the works are many new products we'll be letting you know about soon, if you'd like to take a closer look. Like I/O cards, tape drives, an impact printer--all for the S-100 BUS--and we're designing peripherals for a lot of other CPU's too.

We've given you a little glimpse of who we are and what we're doing. If you want to see more, just fill in the coupon below.

P12

PERIPHERAL VISION INC.

P.O. Box 6267/Denver, Colorado 80206 303/777-4292

Name ______________________________

Address ______________________________

City/State/Zip ______________________________

```
 10 REM :FUNCTION SPEED AND
 20 REM :ACCURACY BENCHMARK
 30 PRINT "START"
 40 LET T=0
 50 LET E=0
 60 FOR I=1 TO 1000
 70   LET S=EXP(LOG(I))
 80   IF ABS(S-I)/I<=T THEN 100
 90     LET T=ABS(S-I)/I
100   LET R=1/I
110   LET F=SQR(SIN(R) ^ 2 + COS(R) ^ 2)
120   IF ABS(F - 1.0)<=E THEN 140
130     LET E=ABS(F - 1.0)
140   NEXT I
150 PRINT "DONE"
160 PRINT "LARGEST ERROR IN EXP, LOG=";T
170 PRINT "LARGEST ERROR IN SIN, COS, SQR=";E
180 END
```

Program 1.

```
 10 REM :STRING MANIPULATION BENCHMARK
 20 PRINT "START"
 30 LET A$="0123456789"
 40 LET B$="ABCDEF"
 50 FOR I=1 TO 1000
 60   LET C$=A$ + B$
 70   LET C$=LEFT$(C$,1)
 80 NEXT I
 90 PRINT "DONE"
100 END
```

Program 2.

EXP(LOG(I))=I and SQR(SIN^2(X) + COS^2(X))=1, this is an easy test to make. It's also a fairly severe test, since you might expect that even if, for example, the LOG function is accurate to six decimal digits and EXP is accurate to six, that EXP(LOG( )) might be accurate only to five places or so.

SWTPC comes out well in the function accuracy test, T=1.0E-06 and E=3.0E-07, which means you can really trust SWTPC's arithmetic functions to six decimal digits. The corresponding error measurements for Sphere are T=.418655 and E=1.6307E-4. That means you can trust Sphere's EXP and LOG to no decimal digits, and you can trust SQR, SIN and COS to three decimal digits. This may be a bit overstated — the real problem is that Sphere's LOG function is worthless for arguments around 1.0. Outside that range, it seems to be accurate to three or four decimal digits.

In terms of speed on the function test, there's a bit of a surprise. Here, SWTPC is in the same ball park as Sphere! For the first seven benchmarks, SWTPC is from 10 to 30 times faster than Sphere. All of a sudden, in the function benchmark, it's less than twice as fast. This puzzled me, so I started playing around.

At first I hypothesized that SWTPC was just using particularly bad algorithms to compute the functions. This began to look likely when I discovered that I could write a program *in BASIC* that could compute the SIN function to nine-digit accuracy almost as fast as the built-in SIN could compute it to six places! (Mine takes 75 seconds to compute the sine of the 100 angles from -49 to +49 radians, while the built-in SIN function takes 72 seconds to do the same.)

Convinced that I was on the right track, I coded my routine in machine language (but using calls to SWTPC's add, multiply and divide routines), expecting to speed things up by a factor of 20 or 30. But my machine-language sine routine runs just 3.5 times faster than SWTPC's. This suggests that most of the time in computing the arithmetic functions is spent in the add, multiply and divide routines, with the overhead of moving the arguments around, looping and testing approaching insignificance. I haven't successfully isolated the problem — every test I've been able to think of has been inconclusive. I *am* suspicious of the use of the BCD representation, however. To get nine decimal digits of accuracy, SWTPC allocates five bytes for the fractional part of each number. Using normal two's complement binary representation, you need just four bytes (2^{31}=2 147 483 648). Not only does using BCD take up more space, but the arithmetic operations are harder to perform.

Each of SWTPC's arithmetic routines is surprisingly long — even the one for negating a value (which is next to trivial using two's complement). Their documentation claims that using BCD gives greater accuracy. Nonsense. TSC (which also uses BCD representation) at least says something true — it's easier to convert decimal numbers into BCD than into two's complement binary. But, so what? BCD makes sense for supermarket cash registers, where the memory required to store the conversion program exceeds the amount of memory needed to store the inefficient BCD numbers, but not in a general-purpose situation.

Ease of Installation

All four BASICs are written for the 6800 microprocessor. Of course that doesn't mean that they'll work properly on just any old 6800-based system. Some are specifically fitted to specific systems (Sphere's is intended to work on the Sphere 330 or 340 and no other machines; SWTPC's is intended to run on the SWTPC 6800 computer and no other; Micro-BASIC Plus is intended to run on a number of different systems; Tiny BASIC is designed to be usable on a very broad range of 6800-based systems). Given your specific system, you face a variety of potential trouble spots in trying to install a given BASIC. To name a few: Do you have RAM at the locations required by the interpreter? Are your I/O interfaces of the type and at the addresses assumed by the person who wrote the interpreter? Does your output device accept and properly interpret the control characters these versions tend to tack onto output? Etc., etc.

The conflict that caused me the most trouble is a battle for the use of the bottom page of RAM — on my system the ROM monitor uses part of the bottom page for temporary storage. With the 6800, unlike with the 8080, there's a legitimate reason for the hardware manufacturer to preempt part of the bottom page of memory — instructions that refer to the bottom page take up one less byte than those that directly access higher memory locations. If a manufacturer provides a large program in ROM, he'll save space, hence money, using the bottom page for storage. Of course, the software producer has to make sure his software package takes less space than the competition's; so, he'll also want to use the bottom page. Everything's fine except for the end user: you and me.

Table 7 shows about how long it took me to get the four versions working on my system. Of course, the numbers for Sphere and SWTPC can change radically, depending on which machine you have. If I hadn't had a Sphere system, the time required to get Sphere provisional nonextended BASIC working would have been substantial.

Note! Please do not write me asking for a copy of my kludged-up version of SWTPC

8K BASIC. You don't want it. It's not the right solution to the problem. Maybe if enough people with 6800-based non-SWTPC machines wrote to SWTPC, they would do something about it.

Pittman does the best job of explaining how to install his version. He even provides hints and sample programs for the necessary I/O drivers on seven different systems.

TSC says in their advertisements that their version requires continuous memory space from location 0 up through the end of everything (i.e., at the end of your BASIC program is space for data values; your program is stored after the interpreter itself). Since that condition isn't satisfied on my system (due to the conflict with locations used by my monitor), maybe I should have just given up there. But, instead, since a well-commented assembly-language source listing of the interpreter is included in TSC's documentation, I was able to go through and shift references to memory locations my monitor uses. That took about four hours.

Then I spent about two hours writing the I/O drivers, and I was ready to watch it fly. I typed in PRINT "HELLO" and was rewarded with HELLO. So far, so good. Then I typed PRINT 2 and got 2. Feeling confident, I typed PRINT +2 and was presented with a situation in which the processor ran wild, wiping out part of the interpreter, winding up in an infinite loop, endlessly cycling through part of high memory space. (Let me skip the gory details of how I was able to detect all that.) I spent some four or five hours tracking down where the problem was occurring (this phase would have taken much longer without the source listing of the interpreter). Finally, two things clicked into place, and I realized what was going on.

First, the problem didn't occur in one specific spot, but randomly in one of three consecutive instructions. Second, the carnage started at a consistent location, which happens to be the place my system jumps to respond to a hardware interrupt. Finally, I looked over the assembly listing and discovered what I might equally well have been able to see in a few minutes if I had just thought of the possibility earlier — certain tests in the TSC arithmetic routines have the side effect of turning off the 6800 interrupt mask! Milliseconds after that, my real-time clock (or *any* of my I/O interfaces) calls for an interrupt, causing the processor to branch to where the interrupt-handling routine should have been — and Kapow! OK . . . so out with the manuals to figure out how to tell all the PIAs and ACIAs not to request interrupts, grumble, grumble, and finally, after about 12 hours, Micro BASIC Plus was up and flying.

I've saved SWTPC 8K for last in this discussion of "ease of installation" because it's a special case. SWTPC wants you to buy an SWTPC 6800 computer to run their BASIC on — they make no claim that it'll work on any other system, and they supply no instructions for installing it on anything but an SWTPC 6800. That means you'd have to be a fool, crazy, or both to try to bring it up on any other system. I plead guilty, and promise I'll never do it again. But since I did it this once, I thought it might be interesting to run through some of the trouble spots.

Actually, it was a team project. My friend Nick did a couple of days' work to get things started. He deciphered the SWTPC cassette-tape format and figured out where to insert calls to I/O routines, then turned it over to me. At this point, we had a BASIC that gave us an error message for every line or command we typed in. It also put out a tremendous number (for my system) of extraneous control characters, so that the ready prompt looked something like:

```
##
####READY
```

But that was a secondary problem.

Armed only with a few old SWTPC newsletters and a lot of coffee, I plunged in. For a few days, I alternated between trying to track down the location where everything went haywire and just browsing through the 7K+ of raw machine code, trying to get some vague feeling for what was what. Let me recount one problem in some detail.

Arithmetic expressions weren't being evaluated properly, so I started tracking *down what* the interpreter did to evaluate them. I established that a particular return address was getting clobbered in the middle of processing the expression 1 + 2*3. Before a particular subroutine jump, the return addresses on the stack were OK. After it, one of them had been changed, changed to a value that was obviously wrong. So, I started looking at the subroutine to see if I could figure out (vaguely) what it did. I couldn't really tell, though, since it consisted of a bunch of subroutine calls. I jumped into the middle, set a break-

Version	Time to install	Comments
Sphere provisional nonextended BASIC	30 minutes	No problems — made to run on my specific system
Pittman Tiny BASIC	3 hours	No particular problem — just had to write I/O routines and test for break character routine.
SWTPC 8K BASIC	75 hours	SWTPC 8K BASIC is made to run on the SWTPC 6800 only. Since I had neither an SWTPC 6800 nor a source listing of the BASIC interpreter, I had to slog through raw machine code trying to find out why it kept blowing up, to find where the I/O drivers were called, locate references to memory locations that my monitor thinks are its alone, change control characters, alter stack pointer initializations, etc.
TSC 6800 Micro BASIC Plus	12 hours	As they say in their catalog, ". . . TSC 6800 programs require RAM starting at memory location 00 and continuing uninterrupted through the amount required by the program . . ." and, to repeat, there are some low memory locations that my monitor thinks it owns. It took me about six hours to go through the source listing to find all conflicts, decide where to move the offending memory assignments, make the changes, write I/O routines. Then it took another four hours or so to discover that TSC turns off the interrupt mask during some arithmetic operations. That's rude of them.

Table 7. Ease of installation.

SWTPC 8K BASIC uses a simulated stack (i.e., a stack that uses neither the 6800 stack pointer register nor the 6800 stack operation instructions) to save the contents of the index register from time to time. Naturally, there is a Push routine and a Pop routine, and a pair of memory locations to store the address of the current stack top. And, of course there's code that initializes the address at the beginning of everything. When installed on the SWTPC 6800 computer (or on any system using the MIKBUG monitor), this address is initialized to $A07F_{16}$. Since I have no memory at that location, I picked an address essentially at random in a region where I *do* have RAM. I picked 4800_{16}. That turned out to be a disastrous choice. Here's why.

The Push routine works like this:

get current stack pointer
decrement it twice (using the DECX instruction, which performs a full 16-bit decrement)
store the contents of X in the place pointed to by the stack pointer
return.

So, after carrying out the Push routine once, the value of the stack pointer will be 4800-2 = $47FE_{16}$. So far so good.

The Pop routine works like this:

get current stack pointer
copy the contents of the (two) bytes indicated by the stack pointer into the X register
increment the second byte of the stack pointer twice (using the INC instruction, which performs an 8-bit increment)
return.

Now. Suppose we've just returned from whatever dirty work we were up to and want to restore the index register to its previous value. We call the Pop routine; it restores the X register and *then leaves the stack pointer with the value 4700!* (Not 4800, as it should be.)

After the next Push/Pop sequence, the stack pointer will be at 4600, and soon it'll be down among our BASIC program, wreaking havoc!

Note that this is *not* a bug in SWTPC 8K BASIC. The author of the interpreter knew what he was doing, had determined that no combination of circumstances could require the use of more than $7E_{16}$ bytes in this particular stack, and so initialized the stack pointer to A07F. The cure for my case was simple. All I needed to do was initialize the stack pointer to 477F instead of 4800. If only I had had some way of knowing that beforehand!

This particular problem is just one of many. It's of no great interest in and of itself — it does show the sort of pitfalls awaiting you when you try to do things the wrong way. No person in his or her right mind would attempt to install a 7000-byte machine-language program on an alien system with no relevant documentation.

Box 2.

point, got back into BASIC, typed in my sample statement (PRINT 1 + 2*3), hit return, and then when the interpreter hit the breakpoint, I looked at the return address that was causing trouble. It was still OK. So, I moved the breakpoint later in the subroutine and repeated until the offending call revealed itself.

Then I looked at *that* subroutine. It too consisted of a bunch of subroutine calls, so I repeated the process. After several hours, I finally found the place where the return address was being clobbered. Box 2 describes what was causing the problem. Once I figured out what was wrong, the solution took about 30 seconds! Finally, I could evaluate arithmetic expressions. After just two days' hard work.

If I had it to do over again, I wouldn't. What I have now is a kludged-together, slightly unstable mess, which usually runs properly. If and when SWTPC issues a new, improved version of their 8K, I'll have to go through hours of work to bring it up on my systems — most of the time I've put in on this version will be wasted. Making absolute machine-language patches in a large program that is virtually undocumented is just the wrong way to go. Period.

Bugs — Theirs and Yours

Let's face it. Sphere provisional nonextended BASIC (the only version Sphere has ever made available, to my knowledge) is a disaster. The others have few blatant bugs, but there are some rough edges here and there. In Micro BASIC Plus, for example, if an array A has been dimensioned to be of size 25 (say), your program can refer to and store into all locations from A(-25) to A(25). Apparently just the absolute value of the subscript is checked. This can lead to some hard-to-discover bugs in programs that involve complex subscript expressions. In addition, if by chance you give an array a dimension of 99 (which the manual says is not legal, but the interpreter doesn't check for), no subscript checking seems to be done at all, thus giving you a method to wipe out other data, your program, the interpreter. . . . Another rough edge in TSC's BASIC is that leading zeros aren't suppressed on printed negative values.

Here's a rough edge in SWTPC's 8K: The test for string inequality is a little wacky if the string values being compared are of different lengths. This makes it a little awkward to get "SMITH" to come before "SMITHY" when you're putting a list of names in alphabetical order. (See Table 8 for a fix you can use that isn't too terribly slow.)

I've used Tiny BASIC frequently and haven't come across anything I'd be willing to call a bug *or* a rough edge.

Of the four, Sphere seems to do the best job of error checking the program as you enter it. Of course, there's no way to know what the error numbers mean since that wasn't included in the documentation. At any rate, Sphere is the only version that checks each newly entered line for syntax errors as it's entered. Micro BASIC Plus will complain if it can't identify the keyword of a newly entered line (but not about any deeper errors). SWTPC doesn't complain about illegal keywords on entry, but does print a question mark in front of them when you list your program. Tiny BASIC doesn't check the newly entered line at all, it just stores it. That's no help in writing programs, but it does mean that you can use the Tiny BASIC system as a sort of text editor — you could enter in some text, list it, correct it, make a hard copy listing, and then cut off the line numbers, I suppose.

All four versions give error diagnostics if an illegal statement is encountered during execution of your program. I ran some tests to judge how appropriate the error messages seemed to be — Tiny BASIC won most of my tests, although I suspect that Sphere would win if I knew what its error numbers meant.

Overall Conclusions

Sphere BASIC has the most bugs in it, and is the hardest to use. Pittman Tiny BASIC is the easiest to install on the widest range of systems (assuming you have some way to read paper tape). SWTPC 8K BASIC has the most features and comes closest to the proposed ANSI standards. TSC's Micro BASIC Plus runs the first five benchmark programs faster than any of the other three.

```
 10 PRINT "NAME 1=";
 20 INPUT S$
 30 PRINT "NAME 2=";
 40 INPUT R$
100 REM :COMPARE THE STRINGS
120 IF S$ <=R$ THEN 200
130 PRINT R$;" COMES BEFORE ";S$
140 PRINT
150 GOTO 10
200 PRINT S$;" COMES BEFORE ";R$
210 PRINT
220 GOTO 10
230 END

RUN
NAME 1=? ADAMS
NAME 2=? BRONSON
ADAMS COMES BEFORE BRONSON

NAME 1=? SMITHY
NAME 2=? SMITH
SMITHY COMES BEFORE SMITH

 10 PRINT "NAME 1=";
 20 INPUT S$
 30 PRINT "NAME 2=";
 40 INPUT R$
 50 IF LEN(S$) <=LEN(R$) THEN 120
 60 REM :R$ IS LONGER, SWAP 'EM
 70 LET T$=R$
 80 LET R$=S$
 90 LET S$=T$
100 REM :COMPARE EQUAL SIZED
110 REM :PARTS OF THE STRINGS
120 IF S$ <=LEFT$(R$,LEN(S$)) THEN 200
130 PRINT R$;" COMES BEFORE ";S$
140 PRINT
150 GOTO 10
200 PRINT S$;" COMES BEFORE ";R$
210 PRINT
220 GOTO 10
230 END

RUN
NAME 1=? ADAMS
NAME 2=? BRONSON
ADAMS COMES BEFORE BRONSON

NAME 1=? SMITHY
NAME 2=? SMITH
SMITH COMES BEFORE SMITHY
```

Table 8. The program at the top shows the effects of the quirk in SWTPC 8K BASIC's test for string inequality. "SMITHY" comes before "SMITH." One possible fix appears in the program at the bottom. Now names will be put into conventional alphabetical order. (Underline = operator input.)

Tiny BASIC requires the least amount of memory, with Micro BASIC Plus a close second.

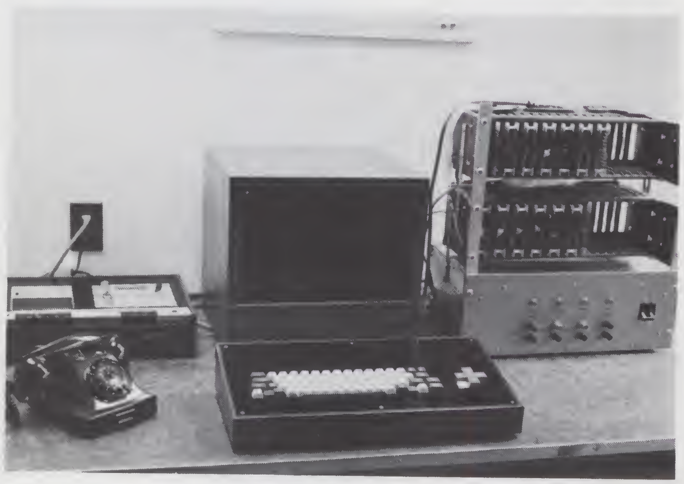
Here's the overall picture, then. Any one of the four might be best, given your specific circumstances . . . well, on second thought, I can't imagine any credible circumstances in which Sphere would come out on top. If you have just 4K of memory, you want Tiny BASIC. If you have 12 or 16K of RAM in your SWTPC 6800 and you want to have a full version with strings, you probably already have SWTPC 8K BASIC. If you have a 6800-based system, which leaves the bottom page of memory alone, you'll want to look into Micro BASIC Plus. If you don't have a system that uses the Motorola MIKBUG monitor system, and you feel apprehensive about grunging around in machine language trying to bring BASIC up, you probably want Tiny BASIC.

And if you insist on having a super-fast, super-cheap, easy-to-install version that will accept any ANSI Minimal BASIC program, well, you'll have to wait. ■

Dr. Michael Wingfield
93 Pine Hill Rd.
Bedford MA 01730

Hardware Program Relocation

the way the biggies do it



Author's home-brew 6800 system.

A common method used within the minicomputer world to achieve better utilization of memory and provide some flexibility to programmers is a system referred to as "memory management" or "memory map." With this system, the programmer can assemble all of his programs in location zero, and a hardware/software combination will determine where in memory the program will be stored. In addition, any program, regardless of its origin, can be relocated anywhere in memory. Michael discusses the hardware for incorporating such a system into a microcomputer. In a subsequent article, he'll take a look at the support software. — John.

Often, one would like to relocate programs in memory without reassembling them each time with a new origin. Programs on programmable read only memories (PROM) could then be relocated without having to burn a new set of chips; programs on diskette could be loaded anywhere in memory, and, for those with more sophisticated systems, programs can be moved in memory to free up space for additional programs.

With a small hardware addition to a microcomputer, programs can be relocated without the need for special assemblers or loaders. Programs can be assembled relative to location zero, loaded anywhere in memory and executed correctly by doing nothing more than setting a hardware parameter.

Base Register Addressing

The general hardware technique for enabling program relocation using a base register has been used for many years in large mainframes. It consists of using hardware to modify the address on the address bus before it is presented to the peripheral devices and memory. By addition of the contents of a hardware base register to the address bus, a new, modified, address is obtained. For example, a program reference to location E061 hex, modified by a base register containing 162A hex now references F68B hex. How is this useful? Consider the following 6800 program:

```
              ORG      0
BUFFER        RMB      1
              .
              .
              .
              LDA A    BUFFER
              .
              .
              .
```

An assembler would assign BUFFER to location 0. This program can be loaded anywhere in memory, say 1000 hex, which places BUFFER at location 1000 hex. With a base-register value of 1000 hex, the load reference to BUFFER would present a value of zero on the address bus, which would be added to 1000 hex to yield 1000 hex, the actual location of BUFFER. Similarly, with the program counter set to 10 hex, the actual byte fetched would be from 1010 hex.

Partial Address Modification

Use of a partial address-modification technique results in a more practical system that requires less hardware. By modification of only the upper eight bits of the address, for example, program relocation on page boundaries can be achieved, implying a page size of 256 bytes. In other words, programs can be loaded starting on any page boundary in memory. Fig. 1 shows the general technique. The lower eight bits of the microprocessor are passed through unchanged. The upper eight bits are added to the base register to form a modified address. Eight bits were chosen because it is a convenient byte size and considerably simplifies the hardware.

Fig. 2 shows the address-modification hardware in more detail. A hardware register and an eight-bit adder combine to modify the upper address byte. The base register is designed to be loaded from the data bus and can be read by the CPU, an important consideration for the software required to support a system with hardware address modification.

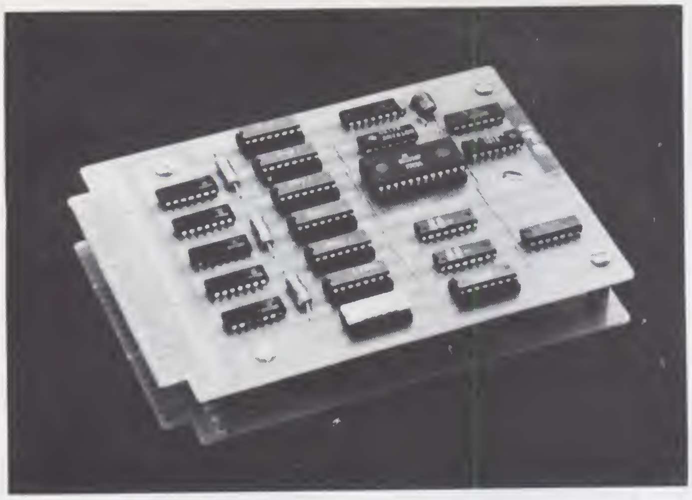


Photo 1a. Top view of board.

Photo 1b. Wiring side of board.

The figure is complete except for one additional piece of hardware that disables the base register and allows the full address to pass through unmodified. Why is this necessary? An absolute piece of system software must be resident to enable changing the base register. If any relocated program stored a byte in the base register, the effective program counter would instantly point to some other place in memory, maybe out of range of the executing program.

Fig. 3 shows the complete hardware necessary to effect program relocation via a base register. The base register is disconnected with eight AND gates, which are enabled by any address in the upper 4K of address space. The upper 4K is used for system software, chosen because it contains the interrupt vectors of the 6800 and is large enough for a reasonable operating system and system stack. Although the control signals reflect the Motorola 6800 system architecture, the logic can be adapted to work with other modern microprocessors.

The base register was assigned a memory location in the upper 4K of memory, in this instance F606 hex. A read of this location obtains the current contents; a write into this location changes the register contents. Connection to the MPU data bus is made with 8T26s, which minimize bus loading.

Interfacing and Debugging

The base-register logic is inserted directly in the address bus of the 6800, between the MPU and any memory or peripheral devices. Although the logic was installed in a home-brew microcomputer, installation in an existing kit such as the SWTP 6800 requires that the eight most significant address wires be cut as they emerge from the CPU board, and the new logic interposed.

Debugging a new microcomputer assembled from CPU, memory and peripheral chips is always a difficult undertaking because of the complexity of the large-scale integrated circuits and the extremely limited means available for determining what the system is doing at a particular time. In general, the developed hardware and software must be nearly perfect, so that after hitting reset, some repetitive function that can be displayed on a scope or terminal is performed. Inserting a base register in the address path complicates the debugging process as the register must be loaded correctly, and correct addresses must be generated by the hardware.

Debugging my system began with a tiny program in PROM that loaded the base register and caused the CPU to loop on itself following reset. Once this was made to work, as evidenced by correct oscilloscope measurements of the address bus, data bus and control signals, more complex software was developed by use of a bootstrapping process. (Bootstrapping is an iterative technique of using currently existing software to develop the next generation of more functional software.)

Debugging problems included the usual software errors and hardware wiring errors, which, in the initial stages of debugging, consumed great quantities of time. Obeying the MPU chip sets interconnection rules exactly, and using careful construction techniques at the beginning of a project can

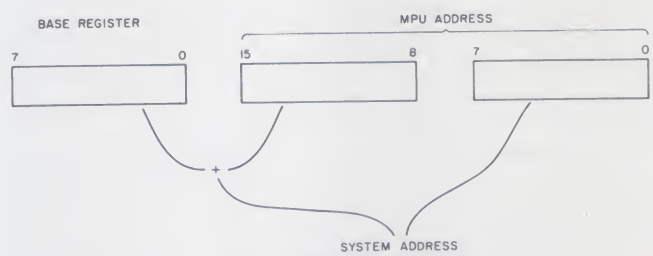


Fig. 1. Address calculation.

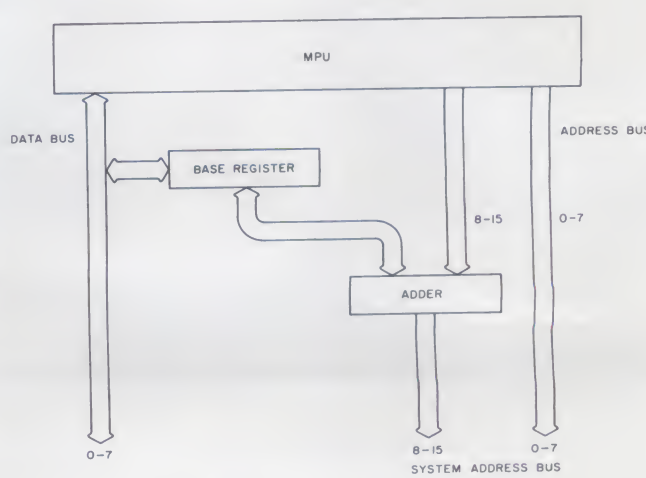


Fig. 2. Address modification technique.

cut a considerable amount of time off the development period.

Prototype System

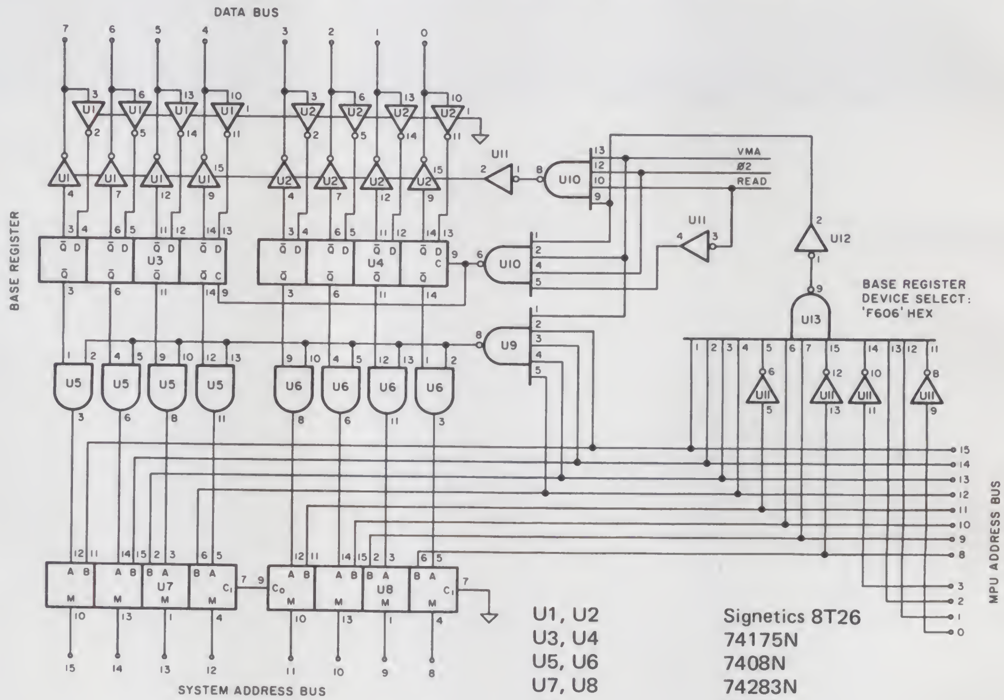
I implemented the base-register logic by using prototype boards that I developed for general breadboarding purposes. Each printed circuit board contains +5 voltage and ground strips and an array of predrilled holes into which all sizes of integrated circuits can be inserted. Connections between ICs are made by soldering Teflon-coated wires directly to IC pins. Photos 1a and 1b show the top and bottom views of the prototype boards, which are usually paired as shown to make more efficient use of board real estate and bus interconnections. Although soldering the wires to the IC pins requires some dexterity the technique produces a very reliable logic system.

Summary

An additional feature gained by use of the address-modification hardware technique with the 6800 processor is most interesting. On an unmodified 6800 system, the direct mode instructions can refer to operands in the first 256 bytes of memory only. With the addition of the address-modification logic described here, a program located anywhere in memory can use direct mode instructions on the first 256 bytes of its program space. For example, a program originating at location zero, loaded at location 1000 hex, with a base register equal to 10 hex, can use the space 1000 to 10FF hex to hold operands for direct mode instructions. This is possible because the processor thinks it is referencing the first page of memory.

This configuration also allows programming in the usual way of originating at any arbitrary address and loading it there. Just set the base register to zero.

By the addition of a small amount of hardware to a microprocessor system freedom from the constraint of absolute memory references can be achieved. The hardware described here has been in use for some time in my system and has proven beneficial. In a subsequent article, software for supporting the base-register architecture will be examined. Together, they provide a degree of programming freedom far beyond that of conventional systems. ■



U1, U2	Signetics 8T26
U3, U4	74175N
U5, U6	7408N
U7, U8	74283N
U9	7430N
U10	7420N
U11, U12	7404N
U13	74S133N

Fig. 3. Base register logic diagram.

Don Lancaster
Synergetics

TVT Hardware Design

low-cost graphics

This is the second of a two-part series by Don Lancaster, taken from his new book The Cheap Video Cookbook *which will soon be published by Howard W. Sams. — John.*

The next most important block in our cheap video interface hardware is the data-to-video converter. The data-to-video converter receives code from the display memory's upstream tap and converts the input data into serial video. The input data can represent ASCII characters, hex op code or graphics chunks.

Graphics data-to-video conversion is usually simpler than alphanumeric conversion. For graphics use, we can sometimes get by with nothing but a shift register that converts the parallel chunk code into serial video. To this we might add an electronic selector to rearrange the chunk as needed for other formats. This selector can be a 4PDT switch that picks the upper or lower part of a chunk on a given scan.

For alphanumeric TVTs, there is no one-on-one relationship between the ASCII and cursor code stored in the display memory and the dots on the screen. Somehow, we have to irrationally fluff up our 6-, 7-, or 8-bit code into a 35- or 63-dot serial-video code. Since the character dots *don't* have any logical relationship to the ASCII code, any bits-and-pieces logic scheme is bound to be a complex disaster.

Instead, we go to the code conversion capabilities of a read only memory (ROM). You can use your own PROM for this, but code-converting read only memories called *dot-matrix character* generators are easy to get, usually cheaper and often a better choice. Details on these character generators appear in the *TV Typewriter Cookbook* (Sams 21313). Character generators can offer a choice of uppercase or combined uppercase and lowercase. They will either do the entire conversion to serial video by themselves, or else they will have multiple outputs that have to go to an external video shift register for final conversion.

For TVT use, your character generator must be of the *row-scan* type. There is another type called a *column-scan* character generator, but this is only good for strip printers, advertising signs and similar uses where the serial or parallel output runs up and down rather than back and forth.

An alphanumeric data-to-video converter using a 2513 character generator is shown in Fig. 12. The character generator accepts ASCII words from the upstream tap on the display memory. These ASCII words change once each microsecond for each new character to be output. The 2513 also accepts three "what line is it" commands from the instruction decoder. In exchange for these inputs, five dots are output at once, corresponding to one row on a 5 x 7 dot-matrix character. An eight-input, one-output shift register then converts these dots, along with spacing "undots" from grounded inputs, into raw serial output video. The input ASCII character coding repeats itself at least seven times to generate the entire seven dot rows involved in a row of characters. Our shift register is driven by a high-frequency timing circuit that outputs a narrow LOAD pulse once each microsecond, along with a CLOCK output that runs continuously at the desired dot rate.

An optional cursor is shown in the lower right of Fig. 12. The 4584 is a five Hertz oscillator that sets up the cursor winking rate. If ASCII input bit 8 is high, the CUR input will go high, and a white line is output on leads 01 through 05. The right diode causes this line to blink off and on, while the left diode allows winking cursors only during valid character times.

Since lowercase is not available, ASCII bit 7 remains unused. Note that if you want to display lowercase characters as uppercase, you must add a simple external gate, for the lower six bits of a lowercase u are the same as a 5, and not a capital U.

The 2513 is cheap and easy to get. The newer single-supply +5 volt versions by General Instruments and others are far easier to use than the old +5, ground -5, -12 versions. This is particularly important since all the

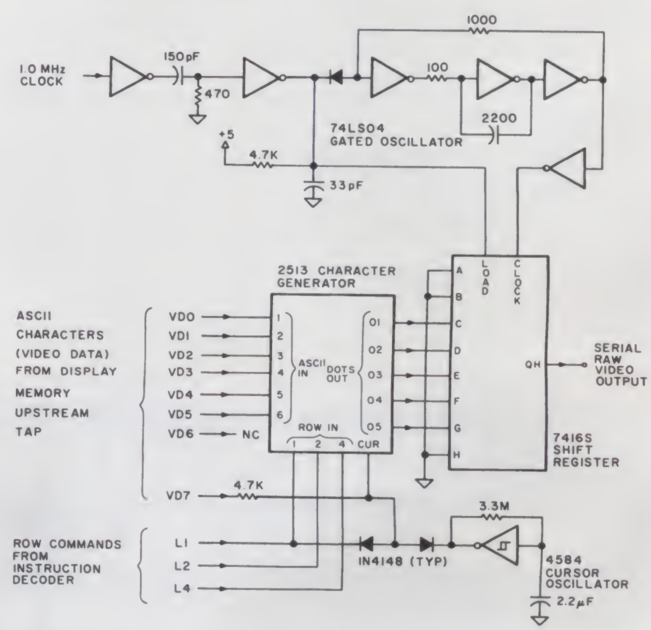


Fig. 12. Alphanumeric data-to-video converter using 2513. Circuit offers uppercase only; uses external video shift register.

rest of the interface hardware can run on a +5 volt supply. Lowercase versions of the 2513 are also available. You can use a pair of 2513s, one upper, one lower, for full-alphabet capability.

A premium 7 x 9 alphanumeric data-to-video converter using the Standard Microsystems CG5004 character generator is shown in Fig. 13. This circuit gives you both uppercase and lowercase and has its own internal shift register. It works on a single +5 volt supply. A winking underline cursor is produced automatically with the cursor circuit shown.

Your turn: Show how switching may be added to either 3-12 or 3-13 to give you manual control of cursor visibility.

Your serial-video output is called *raw video* because it contains only character dots when and where needed and blank logic zeros everywhere else. To get from here to something a TV set, monitor or rf modulator can handle, we have to add the sync pulses, and optionally predistort the raw video for improved clarity. We call everything compensated and combined *composite video*.

Fig. 14 shows us our first graphics interface. This circuit is used in the TVT-6 7/8 whenever eight dots per chunk in a row are needed. Input video data chunks from the display memory are routed directly to the eight inputs of a shift register. High-frequency timing applies just the right LOAD and CLOCK commands to output continuous dots during graphics display times. The pot in the timing is adjusted for minimum overlap or underlap between sequential chunks.

Fig. 15 shows us a graphics interface used for three or four dots per chunk output per row. Upper and lower chunk halves alternate for sequential lines or line pairs. A 74LS258 data selector is added to the inputs to pick upper chunk halves, lower chunk halves or blanking. We've pulled a trick here to introduce blanking. The ENABLE input to the 74LS258 drives all outputs *high* for blanking. To make this point in the circuit have *high* = black, we use an *inverting* data selector, and then pick the *complimentary* output of the shift register. We then end up with an input 1 giving us a white dot, an input 0 giving us a black dot and a blanking command also giving us a black output.

Your turn: Show a low-cost way (jumpers, switches, small module, etc.) to have both Figs. 14 and 15 use the same circuit board. Also provide blanking for Fig. 14.

Color can be added to the circuit in Fig. 15 with an external color modulator. The color format can be three dots on top of three dots, with the remaining two chunk bits letting us call any of four colors plus black.

High Frequency Timing

It's up to the high-frequency timing to give us the LOAD and CLOCK signals needed for serial output of video from the data-to-video converter. Traditionally, these circuits use crystal oscillators and counter-dividers for this job. For many microprocessor-based video display circuits, all we really have to do is use

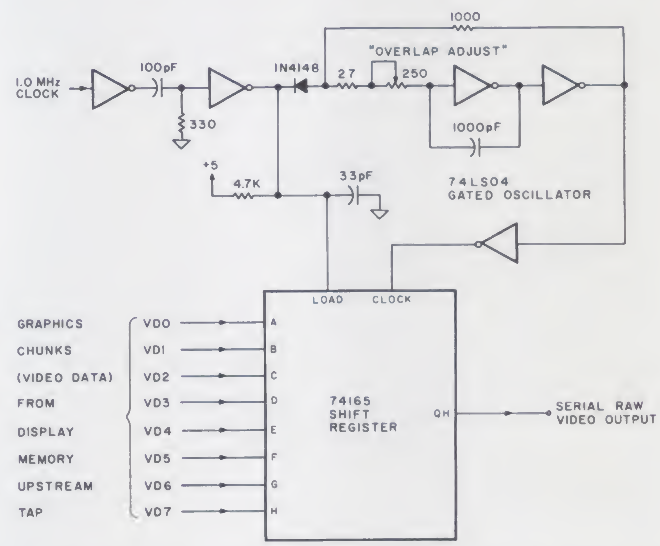


Fig. 14. Graphics data-to-video converter to display eight horizontal dots per chunk.

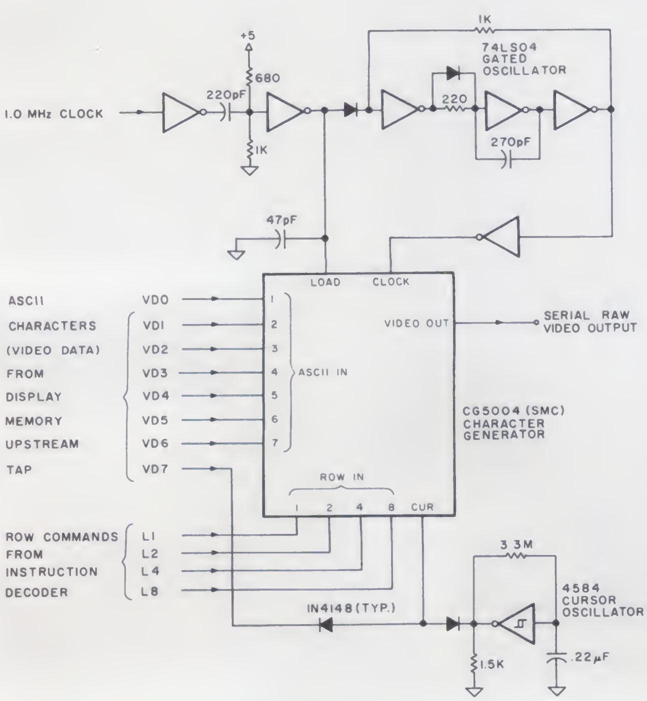


Fig. 13. Alphanumeric data-to-video converter using 5004. Circuit offers uppercase and lowercase; has internal video shift register.

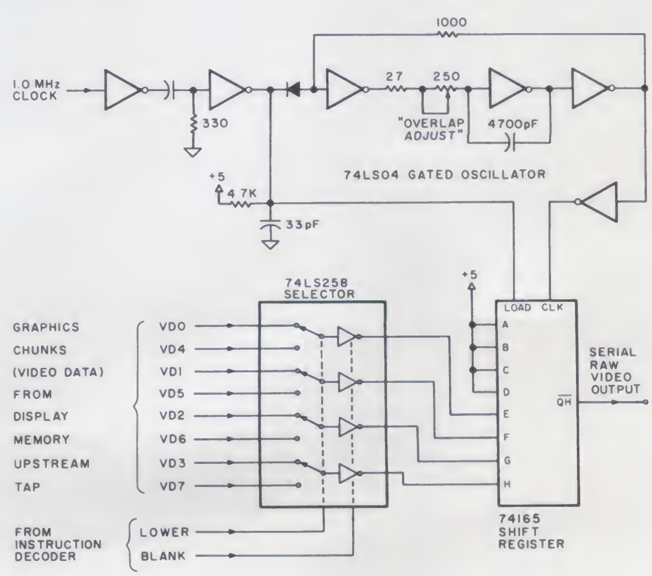


Fig. 15. Graphics data-to-video converter to display three or four horizontal dots per chunk on alternate line pairs.

a plain old one-megahertz microprocessor clock and add a simple gated oscillator using a hex inverter to get our LOAD and CLOCK waveforms.

Four examples of these hex-inverter high-frequency timing circuits appeared on the tops of Figs. 12 through 15. Fig. 16 tells us what these timing circuits have to do, while Fig. 17 shows us typical waveforms.

The LOAD output of the timing circuit has to transfer parallel dots into the video shift register. *This LOAD pulse must be carefully timed to arrive only when data is ready and settled from earlier portions of the data-to-video converter*. Often, you will have two choices of one-megahertz clock available; if one phase doesn't do it, the other one probably will. Most often, it's best to arrange the load command so it always arrives one microsecond after the address change. This gives you a nearly maximum processing time and minimizes any settling or bad-data problems. Note that the cursor and blanking should be introduced *before* this one microsecond delay takes place; otherwise they will also have to be delayed somehow.

The polarity of the LOAD output has to match whatever is doing the final serial conversion. A 74165 shift register needs a normally high LOAD command that briefly goes low for loading. The CG5004 needs the opposite. In general, LOAD command pulses should be as narrow as possible. This is particularly true in graphics interfaces where a too long or misplaced load pulse can cause dot overlap or underlap.

the

HIGH
FREQUENCY
TIMING

must

* Deliver a "load" pulse to the video shift register.
* Time the "load" pulse to arrive only when data is valid.
* Deliver "clock" pulses to set the video dot rate.
* Lock "load" and "clock" pulses together to prevent jitter.

Fig. 16. High-frequency timing traditionally has used crystals and divider chains, but a hex-inverter gated oscillator is often all that is really needed.

Your CLOCK line decides how fast the dots are going to come out as serial video. In an alphanumeric display, too low a dot rate will cause the characters to overlap, while too high a dot rate will make the characters narrow and will also need too much video bandwidth. Between these limits, the frequency of the clock gives us control of the spacing between characters. The CG5004 is very fussy over its clock duty cycle. It demands the narrow positive pulses shown in Fig. 17b.

In a graphics display, the dot rate sets how many dots you get per chunk of input data. Fine tuning the dot rate is usually necessary so that the end dot of one chunk appears the same size as the start dot of the next chunk. A frequency error here will cause dot overlap or underlap.

CLOCK and LOAD must be locked together to prevent the dot locations from jittering or otherwise smearing. It's also particularly important to make sure the LOAD command does not distort the clock on graphics displays; otherwise clockings end up wider or narrower with respect to each other.

The top half of Fig. 12 shows a typical hex-inverter high-frequency timing circuit. The first inverter acts as a buffer to make us independent of system clock rise and fall times. The second inverter is a half monostable, whose output briefly drops to ground for 60 nanoseconds on the falling edge of the clock input. The three upper-right inverters are a gated ring oscillator running at a seven-megahertz rate. Frequency is set by the 100 Ohm resistor and 2200 pF capacitor. Gating is done by the diode. This synchronizes the oscillator to the LOAD command every time the LOAD command goes low. A final buffer and inverter are used to square up and invert the clock line. The 4.7k and 33 pF network is a glitch filter for added stability.

The high-frequency timing circuits are about the same, differing only in speed and polarity details.

Most often, some cut-and-try is needed in getting a hex-inverter high-frequency timing circuit to work the first time. But, once you have a working circuit, it is usually tolerant of normal production component spreads. Fine tuning is usually needed in graphics applications to control overlap and underlap, besides giving a handy way to change from three to four dots per chunk. Usually fine tuning can be omitted on alphanumeric uses as the main effect of slight frequency changes is a small change in character spacing.

Your turn: Show a way of raising and lowering the microprocessor clock frequency to allow locking of the video display's vertical rate to the power line. Can a

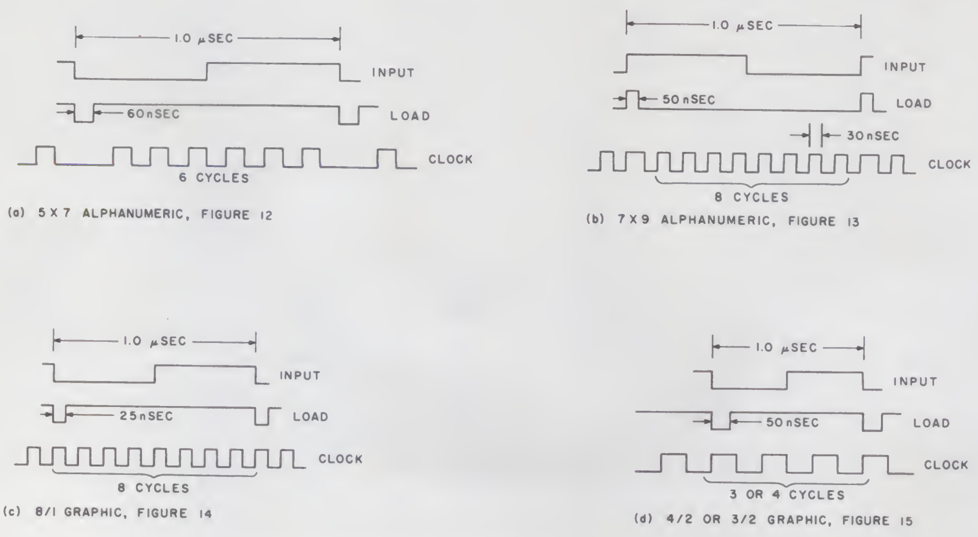


Fig. 17. Typical high-frequency timing waveforms.

CMOS 4046 be of help here?

As a rough rule of thumb, the output video frequency will be around one-half the dot rate set by the high-frequency timing clock. Thus, eight dots per microsecond gives around four megahertz bandwidth, while three dots per microsecond needs only a megahertz-and-a-half bandwidth. A black-and-white television set has a four megahertz video bandwidth, extendable somewhat by defeating the sound trap. The video bandwidth of a color set is limited to three megahertz. As you can see, the output frequencies associated with microprocessor-based video displays are compatible with TV sets with largely unmodified video bandwidths. This is a dramatic improvement over the 10- to 20-megahertz bandwidth often needed by traditional video terminal systems.

Sync and Position

When we run a properly debugged scan program, our instruction decoder will give us signals at the needed vertical (V SYNC) and horizontal (DEN) rates. We can then selectively delay these signals to gain control over position. This is followed by pulse shaping to get the proper widths of the sync signals for TV use. Since the TV set locks itself to the sync pulses, backing up or moving these pulses forward has the effect of moving the entire display. Horizontal delay changes cause back-and-forth position changes in the display, while vertical sync changes control up-and-down positioning.

A typical circuit is shown in Fig. 18. Once again, it is six inverters to the rescue. However, this time two of the inverters have to have very low open collector output impedances and are TTL, while four of them need extremely high input impedances and a snap action, so they are CMOS Schmitt inverters.

Our V sync pulse starts out as a one microsecond positive pulse that is glitch-filtered to get rid of anything that crops up during the PROM settling times. The 7405 discharges the 4700 pF capacitor completely once each 1/60th second. This capacitor is recharged by a rate you set with the V POS control. When the recharging reaches around one-half the supply voltage, the Schmitt snaps on, and our second stage output becomes a square wave delayed by the amount set on the position pot. Because of the extreme differences in charge-to-discharge times of the capacitor doing the positioning, the TTL/CMOS combination is called for. The delayed output is shaped into a positive-going 200 microsecond pulse by the final Schmitt and RC network. The output resistor aids in interfacing the TTL stage that follows in the video output circuitry.

The horizontal circuit is similar, with only the timing details changing. The DEN output of the instruction decoder can often be used, instead of needing a special H SYNC line. Delay of a portion of the horizontal line is done with the first variable RC network, while the second RC combination gives us a five microsecond sync pulse once every horizontal line.

This particular sync and position circuit needs continuous arrival of H and V signals from the instruction decoder. This continuous need limits the transparency and throughput of the computer on other programs that are also active while the TVT is displaying.

Fig. 19 shows us a different way to get horizontal sync pulses. This counter method can free the computer for other uses during vertical retrace times. This in turn can greatly increase the transparency and throughput.

What we have is a divide-by-H counter. H is set to the number of microseconds per horizontal line. For every overflow, an H sync pulse is delivered, regardless of what the computer happens to be doing. This counter is synchronized to the scan program by resetting it with the instruction decoder. The synchronization and reset can take place on every active line, once during vertical retrace, or even only once during power up. Horizontal sync is maintained through vertical retrace times, even if the computer is busy working on something else.

Bandwidth Compensation and Video Output

We now have three signals available — raw video, vertical sync and horizontal sync. We somehow have to combine these and pick up some line

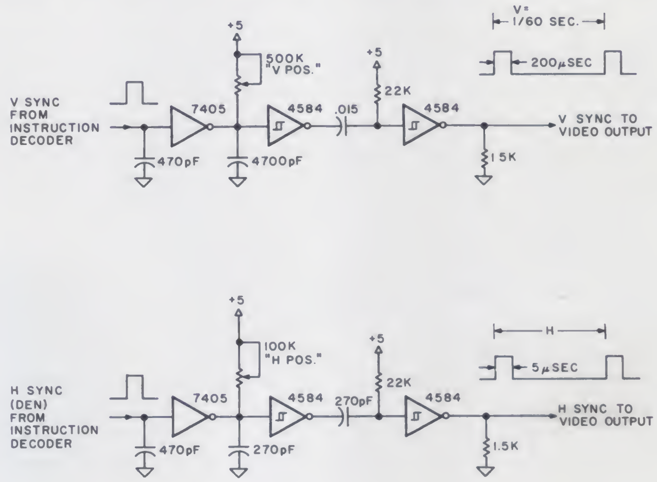


Fig. 18. Sync and position circuitry.

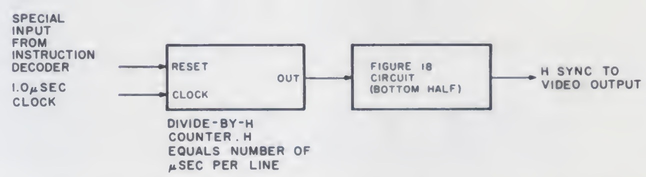


Fig. 19. Modified horizontal sync and position circuit gives high transparency and throughput.

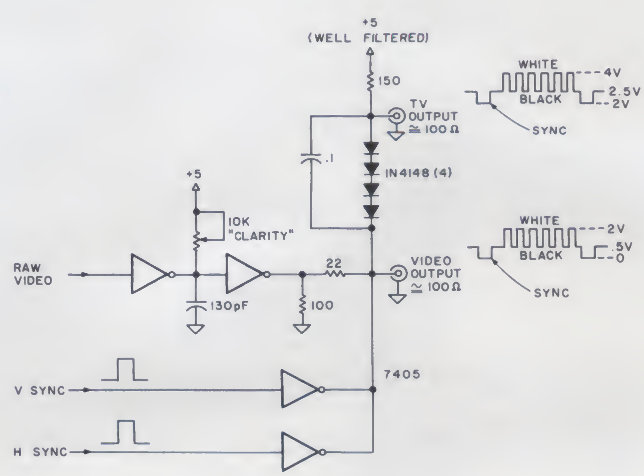


Fig. 20. Bandwidth compensator and video output circuit. Video output is for monitors and rf modulators. TV output is pretranslated for minimum set modifications.

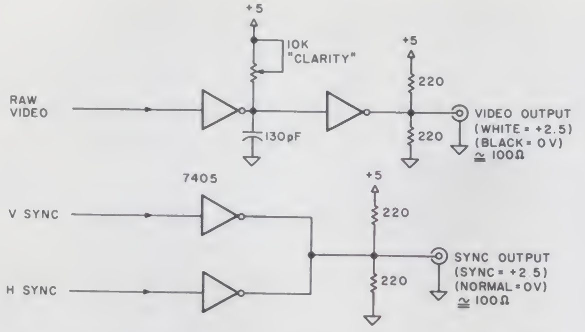


Fig. 21. Alternate bandwidth compensator and video output circuit for monitors with separate sync input.

drive capability if we are going to interface a TV set, monitor or rf modulator. This final interface is the purpose of the video output circuitry.

Our raw video first goes to a *bandwidth compensator*. This super-important circuit tries to anticipate how the TV set is going to degrade the response, and then predistorts the video in the opposite direction beforehand. You do bandwidth compensation by making the dots longer than the undots. One way is to OR the raw video with a delayed replica of itself. A simpler but very sneaky way is shown in Fig. 20. An open collector TTL inverter has a much lower output low ON impedance than its output high OFF impedance. If we add capacitance from this output to ground, the capacitor will discharge fast, but its charge rate will be much slower and set by the value of the pullup resistor, which in this case is a CLARITY pot. Since this is an inverter, a white dot is low and a black undot is high at the capacitor. It takes longer to get out of the low state, so our dots automatically get lengthened.

The amount of lengthening is set by the CLARITY pot. This pot is adjusted for the densest, clearest characters on the final TV screen. The optimum setting is often the one that just barely closes the inside of an M or a W on the display. The use of this bandwidth compensator and our one microsecond constant character or chunk time are the two keys to display of quality characters or graphics on a TV set with unmodified video bandwidth.

Three more open collector inverters are used for video combination. At the VIDEO output, sync pulses are nearly at ground, while black is at 0.5 volts and white is near +2 volts. The ratios of these three values are set by the three resistors. This output is essentially a standard form for video monitors, rf modulators and TV sets that have been completely converted internally for direct video monitor use.

But, we've also provided a new TV output, which has the same waveform but is translated up so that white is at +4 volts, black at +2.5 volts and sync at +2 volts. The +4 white level is the normal bias level at the video detector of most solid-state TV sets. You can often use this TV output to go directly into the first video stage of many TV sets, without needing anything else in the way of translation or bias circuits.

Some monitors have separate VIDEO and SYNC inputs. These are called split sync systems, and an alternate dual output circuit shown in Fig. 21 may be used if split sync is needed or wanted. ■

T. W. Parsons
Polytechnic Institute of New York
333 Jay Street
Brooklyn NY 11201

Understanding Loaders

bootstrap, absolute, relocatable, etc.

Computers, as we know, can do nothing for themselves. Everything they do requires that a program be loaded into them. If you are writing a program, you start by loading an *editor*. If you are translating the program, you must load a translator (an *assembler* or *compiler*). Worst of all, if you merely want to load a program, you must load a *loader*. What's going on here? If you have to load a loader, what loads the loader that loads the loader? Read on.

Of course, I've exaggerated a little. You don't really *have* to load a loader. It's perfectly possible to load a program through the front-panel switches. Just set up the address of the program on the switches, press Examine to enter the address, then set up the first byte and press Deposit (or Store, or whatever your machine calls it) to store that byte. Most machines have a Deposit Next switch, which saves you the trouble of entering addresses after the first one; from then on you just set up each new byte and press Deposit Next to enter it. Nothing to it. Furthermore, some machines (the SWTPC 6800 for example) permit you to load programs just by typing them out on your terminal.

The only problem with this process is that many programs are hundreds, even thousands, of bytes long. Even if you can use your terminal, loading a thousand bytes can be, to put it mildly, tedious. It leads to a feeling that there must be an easier way to do this. Let's see . . . it's a long, monotonous, repetitive job that requires a lot of patience but not much in the way of brains . . . just the job for a computer! And this is the reason loaders are written.

In this article, I am going to discuss loaders in general, and then take up three specific kinds: bootstrap, binary, and relocatable loaders.

Loaders in General

Every loader starts with a copy of a program stored on some external medium, such as punched tape, and the basic job of any loader is to find out where each byte goes, and to put it there. In fact, that's what we did when we loaded the program from the front panel. We had a copy of the program stored on an external medium (for example, a listing typed on a piece of paper or printed on a page of *Kilobaud*), and we did the same two steps — we entered the address first, then set up the byte on the switches and stored it. We knew where the byte was supposed to go because the address was given to us in the listing.

It's the same for loaders. For the simplest loaders, the punched tape contains an address, then a byte of data, then another address, and so on. A special symbol is set aside as an END indicator, and when the loader reads this, it stops.

The data on the tape must be arranged to match the requirements of the loader. For this loader, the punched-tape data format is shown in Fig. 1. Notice that we have two characters of address for every character of data. That's because most micros have 8-bit data words, but 16-bit addresses. (I am assuming eight bits per tape character. If you use 5-bit tape, the details are different, but the principle is the same.) The address has to take up two tape characters because there aren't enough bits in a single character. I am using ADDR(1) to represent the high-order half of the address and ADDR(2) for the low-order half.

The program logic for this simple loader is shown in the flowchart in Fig. 2. The first character read from the tape could be either ADDR(1) or the END symbol, so the program has to make a test to see which it is. We will use hex FF as our end symbol and just hope we never need an address that starts that way. If the first character is hex FF, the program stops. Otherwise, it moves the byte into

the high-order half of the address register. (For example, in the 8080 this might be done by a MOV H D.) The next character, ADDR(2), goes into the other half of the address register. We now have the entire address available, and when we read the DATA byte, the program can store it as required.

Not a very inspired program, is it? First of all, there is no error-checking capability. If your tape reader hiccups while the loader is running, you'll never know until you try to run the program you have loaded. Second, using FF as an END symbol cuts into our address space. We might want to load a program into memory starting at location FF80 sometime. But as soon as the loader reads the beginning of FF80, it will interpret FF as END and stop. Finally, the way we represent the program is very inefficient. For every byte of data, we need two bytes of address. Since most program instructions go into consecutive memory locations, most of these addresses are unnecessary. A smart loader would simply add one to the old address, just the way a person using the front-panel switches would use Depost Next. Still, we have succeeded in mechanizing the loading process, which is what we set out to do.

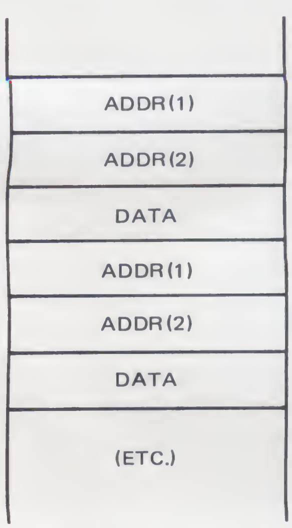


Fig. 1. Arrangement of characters on a punched tape for use with a simple-minded loader.

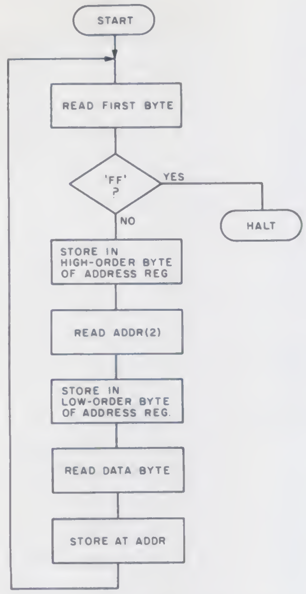


Fig. 2. Logic of the simple-minded loader for use with the tape of Fig. 1.

Bootstrap Loaders

We are now almost ready to answer the question, "What loads the loader that loads the loader?" Grungy as our program is, it does have one important advantage: it is short. It can be written in a mere handful of bytes, which means it's easy to key in. This simple loader can then be used to load in a more sophisticated loader that is too long to key in. Using a little loader to load a big loader is called bootstrapping, and a loader of the sort we have been describing is called *bootstrap loader*. In some machines, the bootstrap loader is put in a ROM inside the machine and, thus, never has to be keyed in. (In the old days, before microcomputers, this feature was called an autoloader, and it cost a bundle.)

All bootstrap loaders aren't the same. Some of them increment their own addresses, some allow you to decide where the main loader is to go. Bootstrap loaders for use with magnetic tape or disks can be very short, sometimes only a couple of instructions. If the main loader is the first record on tape or disk, then the bootstrap loader may require only a single READ instruction. In one refinement of this idea, the instruction is followed by

```
HERE: JMP HERE
```

an instruction which is designed to be *overwritten* by the taped loader. While the tape is being read, the computer keeps jumping to HERE; when this instruction is overwritten, it is replaced by a jump to the beginning of the loader. This makes the loader self-starting.

Absolute Loaders

The more sophisticated loader that is loaded in by the bootstrap is generally called an *absolute loader*. "Absolute" means that it is used for loading programs that are not relocatable — more about that in the next section.

A program tape to be loaded by an absolute loader is organized into blocks. (This is done by the translator when it punches the paper tape.) In front of each block there is a *starting address*, to tell the loader where to start loading it, and a *word count*, so the loader knows how long the block will be. There is also a *check byte*, used for error checking. A typical check byte is computed so that if you simply add up all the bytes in the block, including the check byte, the total (known as the *checksum*) comes out zero. Then the loader can check each block as it goes, and if it ever finds a nonzero checksum, it will write an error message and halt.

Finally, we have to decide on an END indicator that doesn't conflict with our other symbols. One solution is to include a byte that says what *type* of block this is. This is useful anyway, for we will see that more elaborate loaders require several different block types. For the time being, all we will need is a DATA block and an END block.

This brings us to the data format shown in Fig. 3. TYPE comes first, since the loader has to know what type of block it's looking at to know what to do with it. Next come the word count and the starting address. Some people put the check byte before the data, some after.

A flowchart of an absolute loader is given in Fig. 4. For those of you who would rather read programs than flowcharts, I've also sketched out the logic in BASIC in Fig. 5 (Program listing). I don't mean to suggest that this program would actually be a practical loader — for one thing, loaders are *almost always* written in assembly language — but BASIC is a well-known, easily understood language and therefore handy for showing program logic. To relate the flowchart to the program, I've marked the boxes in the flowchart with the corresponding BASIC statement numbers.

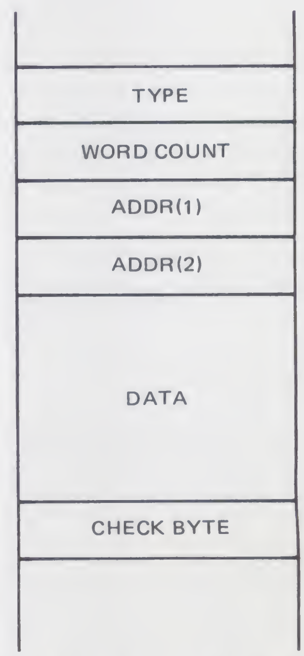


Fig. 3. Tape format for use with an absolute loader.

The first thing the loader reads is the block type. I've decided to indicate an END block by a negative number. Hence, in statement 40 we

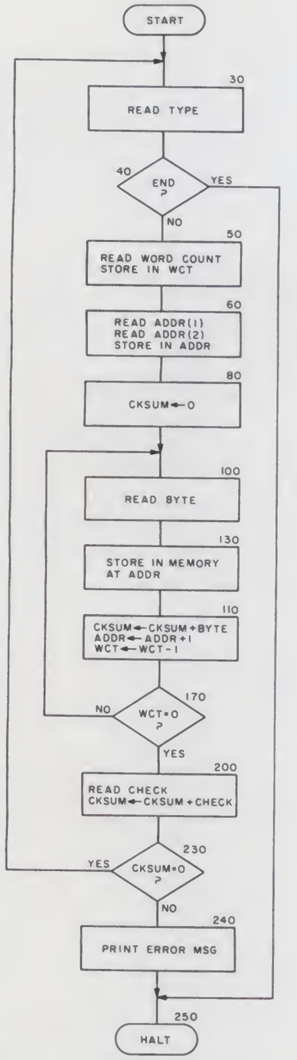


Fig. 4. Flowchart of an absolute loader. The numbers refer to the statements in Fig. 5.

test the sign of TYPE, and if TYPE is negative, we're done. Otherwise, we read and store the word count and the starting address, and zero the checksum. We then enter a loop, which runs from statement 90 to 160. In the loop, we read a data byte, store it in memory and also add it to the checksum. (I've shown memory as a big array, named MEMORY and indexed by ADDR. This would create all kinds of problems in BASIC, but it's pretty close to what's done in assembly language. Remember, the BASIC program is intended just to sketch out the logic.) Then we advance ADDR, so the next byte will be stored in the next address, and decrement the word count. This loop continues until the word count reaches zero. Then we read the check byte and add it to the checksum. If the result is zero, everything is all right, and we go back to the beginning of the program to do the next record.

Location	Contents	Program
0000		START
.		.
.		.
.		.
0030	C37F00	JMP PT2
.		.
.		.
.		.
007F	53	PT2: MOV D,E

Example 1.

There you have the basic design of any loader. Of course, there are variations. Frequently the word count is a negative number that gets bumped up to zero; there may be different data formats; some loaders provide for a jump to a starting point (usually specified in the END block) to make the loaded program self-starting. Some loaders have address checks that make it impossible for the loader to wipe itself out. This is useful in machines with core memory, in which the loader often has a permanent home somewhere in memory, but it is less important these days when a loader can be made bombproof, if desired, simply by burning it into a ROM.

Some loaders work only with magnetic tape or floppies. The logic can be the same, except that the loader will read an entire record from the tape or disk and put it into a buffer area in memory reserved for the purpose.

It then scans through the buffer, using the same program as before, except that where we read characters before, we now just move bytes in from the buffer. For example, statement 30 in Fig. 5 would be changed to read

```
30 LET TYPE = BUFF (1)
```

and the other READ statements would be similarly modified.

Relocatable Loaders

So far, all the loaders we have considered put a program into only one specified region of memory, determined by the addresses contained in the DATA blocks. It is often more convenient to be able to load a program into *any* handy region of memory. This is particularly true for frequently used subroutines, like I/O handlers, mathematical subroutines — or utilities like sorting programs. Programs that can be loaded anywhere are called *relocatable* programs; programs that are not relocatable are called *absolute*. In general (there are exceptions), programs that are put together as required from a library of subroutines are relocatable. Programs that have been put together once and for all by someone else and are not subject to user modification are apt to be absolute — for example, editors, translators, and BASIC interpreters.

Relocatable programs make extra work for both the translator and the loader. The extra work comes from two things. First, if the program contains addresses, the loader has to change all of these addresses to conform to the program's current location in memory. To see why this is so, let's look at the program fragment in Example 1.

In the jump, 7F00 points to the location of PT2, which is given as 007F. Suppose we load this program starting at location 8000. Then PT2 will be at location 807F. If the loader doesn't correct the address in the jump instruction, the program will jump — not to 807F, but to 007F, which might be anything but certainly isn't PT2.

Incidentally, how big a job this address correction is depends in part on the way the computer is designed. Some architectures lend themselves to relocation better than others. In particular, machines that normally compute addresses as displacements from the address of the current instruction, or from some other pointer that can easily be set at the start of

```
10 REM PROGRAM LOGIC OF TYPICAL ABSOLUTE LOADER
20 REM 1. PRELIMINARIES
30 READ TYPE
40 IF TYPE <0 THEN GO TO 250
50 READ WCT
60 READ ADDR(1)
70 READ ADDR(2)
80 LET CKSUM = 0
90 REM 2. MAIN LOOP
100 READ BYTE
110 LET CKSUM = CKSUM + BYTE
120 REM NOW STORE BYTE IN ADDRESS GIVEN BY 'ADDR'
130 LET MEMORY (ADDR) = BYTE
140 REM NOW BUMP ADDR & DECREMENT WORD COUNT
150 LET ADDR = ADDR + 1
160 LET WCT = WCT - 1
170 IF WCT > 0 THEN GO TO 100
180 REM END OF MAIN LOOP
190 REM 3. NOW READ CHECK BYTE & TEST CHECKSUM
200 READ CHECK
210 LET CKSUM = CKSUM + CHECK
220 REM IF CKSUM OK THEN START ON NEXT RECORD
230 IF CKSUM = 0 THEN GO TO 30
240 PRINT 'ERROR!!'
250 STOP
260 END
```

Fig. 5. Program logic of an absolute loader expressed by means of BASIC statements.

the program, require very few absolute addresses, and their programs are readily relocated. (There is more to a machine's architecture than just its instruction set.)

The other source of work for the translator and loader is the fact that if a program contains a call to an external subroutine, the translator won't know where this subroutine is going to be. The call is going to be a JSR (or JMS or CALL) instruction containing the address of the subroutine. The address referred to is called the *entry point*. The translator will have to leave the address blank and leave a message for the loader to fix up that blank once the subroutine in question has been loaded and its entry point is known.

Now, the only means of communication between the translator and the loader is the punched tape, and — yes, you guessed it — we have additional block types to take care of these things. We used to have just DATA and END blocks. Now we will use ADDR blocks, containing a table of addresses that must be corrected, EXT blocks giving a table of external addresses to be supplied by the loader, and ENT blocks giving the location of entry points in the program. These tables have to be made up by the translator, and the translator usually writes the ADDR, ENT and EXT blocks just after the last DATA block, and before the END block.

With all these new blocks, the tape of our program now looks like Fig. 6. How does a relocatable loader (which, come to think of it, should really be called a relocat*ing* loader) handle all this?

The loader will normally load each program into memory starting right after the last program loaded. When it starts to load a program, it remembers the location of the beginning of the program. Let's call this the load address. (In our example, the load address was 8000.) When it processes a DATA block, the loader adds the load address to the starting address of the block, so the contents of the block

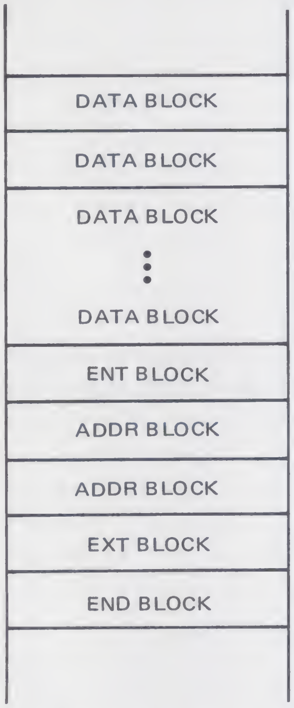


Fig. 6. Sequence of blocks on a tape for use with a relocatable loader.

get stored in the right place. When it processes an ADDR block, it uses the table of addresses to find each address in the program. Then it corrects each address by adding the load address to it.

The fun begins when the loader handles the entry points and external references. It does this by maintaining a table in memory, usually called the External Symbol Table. This table is arranged as shown in Fig. 7.

The first time the loader finds any symbol, it enters its name in the table. If the symbol appears in an EXT block, that means the symbol is an external reference; so the loader copies the address from the EXT block, corrects it by adding in the load address (remember?) and puts it into the References column of the table. If the name is found in an ENT block, it puts the address, similarly corrected, into the Definition column. As more references to the symbol are found in the EXT blocks of other programs, they are also entered in the References column. In this way, a table is gradually built up showing where every entry point is located and where all references to it are to be found.

When loading is complete, the loader goes through the table, and for each symbol copies its definition (that is, its true address) into all the addresses given in the References column. (For example, in Fig. 7, the symbol MULT might be the name of a multiplication routine. Its entry point is located at 06F0, and the loader will copy the address 06F0 into addresses 0405, 0730 and 1C66.) In this way, all of the external symbols are, as we say, resolved. (This process is also known as *linking*.) The loader usually prints a memory map, giving the load addresses and entry points of all the programs. If any symbol turns out to be undefined, the loader indicates an error and prints a list of unresolved symbols.

Like absolute loaders, relocatable loaders often have added features and capabilities, some of them pretty ambitious. In fact, if we follow these developments very far, they will quickly take us into the land of the giants — big systems with dozens of disks and dozens of users — and out of the scope of this article. So, by way of conclusion, I will just touch superficially on some of these possibilities.

Many systems include a library of frequently used subroutines, and usually in these systems the loader will automatically search the library for any programs that are still missing after all the user's programs for a particular job have been loaded. If there is more than one library, you can specify which is to be used and can give other instructions as well. Some loaders have an *overlay* capability that lets you run programs that are too big for memory. You can do this if you can divide your program into parts that don't all have to be in memory at the same time. Each part is called a *phase*, and when your program runs, Phase 1 is loaded initially. When Phase 1 is done, the system replaces it (overlays it) with Phase 2 — and so on. (Working variables are stored someplace where they won't be wiped out.) Notice that here the loading operation begins to be taken over by the operating system itself. In a multiprogramming system, loading may become merely a function of the operating system, and the whole process, including relocation and linking, may be done in mid-execution. The ultimate development of the overlay concept is "virtual memory," in which every user is allowed unlimited memory, and the system swaps parts of his program as needed between disk and memory in order to support this illusion.

This completes our survey of loaders. As you can see, each type of loader has evolved in response to a need. So far, the needs of most hobbyists are fairly modest, mainly because their system resources are fairly modest. As this changes, however, and the use of disk (or maybe bubble) storage becomes widespread, software libraries are going to proliferate, and many of the features we now associate with large systems, such as overlay, are going to appear in hobby systems — although I don't think we are going to be using virtual memory very soon. ■

Name	Definition	References
MULT	06F0	0405, 0730, 1C66
PRINT	0807	0915, 0A17
SORT	10B6	0319, 0AE3

Fig. 7. An external symbol table.

KILOBAUD KLASSROOM NO. 8

pulses – and more pulses

George Young
Sierra High School
Tollhouse CA 93667

In this session, we will take a look at the chips that make up the support chips and functions for computers. In each of the remaining sessions of the series, we will be concentrating on this aspect of the computer. Now, we will study pulse generation and shaping, one-shots, "cheap shots," pulse delay and timing diagrams; and we will build a test unit that can be incorporated into the Design Console or Student Console to generate hexadecimal numbers with the push of a button.

Introduction

Pulses, pulses, pulses . . . that's what computers are made of. Well, that's not quite right. Computers are made of components — called hardware — that work with pulses.

We will look at these pulses that make the computer do its stuff. To do this we must use an oscilloscope. (I mentioned it in previous sessions — and now we're going to use it!) If you have a scope and know how to use it, great! If not, don't run out and buy one since you don't even know what kind to buy yet. Grab that *resource person*; we now need him and his scope. Here are a few possible sources: ham radio operators, fellow computer enthusiasts, computer clubs, local computer stores (using, not buying), local high schools (science/electronics classes), local junior colleges. If you are as lucky as one reader, you might even find a retired electronics engineer living next door.

Experiment #35
The Pulse Waveform

Problem: What do pulses look like?

Solution: Look at them on an oscilloscope.

Procedure: Use the console clock generator and its output waveform as a starting point. Photo 1 shows the pulse waveform for the output of the clock generator in the Design Console. I didn't have a special camera that fits over the viewing screen of a scope, so I just focused a regular camera on the traces and took the pictures. If the results are less than professional, that explains why. You can use Photo 1 as a guide to what to look for on your scope.

Using the Scope

Have your resource person set up the scope for you, and ask him to go through each step slowly while you watch. Ask questions about any step you do not understand. If you comprehend how to do it the first time, then try it yourself. Have the resource person turn off the scope and disconnect it. Then set it up yourself and adjust it so you get the same results. Your resource person should watch as you set things up so he can correct any mistakes you might make.

It may take several trips through the setup procedure before you get the knack of things. Take your time and ask lots of questions. The scope may be simple or elab-

orate, but I know of no other way for you to learn than using one under the supervision of someone who is proficient. It is not my intention here to teach you the scope's inner workings. We want to learn how to use one to look inside an electronic circuit.

After you have a waveform on the scope, vary the speed controls on the console clock and observe the changes produced in the display.

Experiment #36 The One-Shot or Monostable Multivibrator

Problem: What is a one-shot and what does it do? Why is it used?

Solution: Set up the circuit in Fig. 1a on the console breadboard. The chip used for this experiment may be a 74121, 74122 or 74123, all members of a family of monostable multivibrators. Pin-outs for all three chips are given in Fig. 1.

Procedure: Connect the scope to the Q output of the one-shot. If you are using a triggered scope with a separate trigger input, trigger it with the output of the console clock. Then move the scope probe to the $\overline{Q}$ output and again observe the output of the one-shot.

Theory: Note that these chips have two (or more) inputs. One input we will call A, the other B. The A input is a negative-edge triggered input. That is, this input triggers the one-shot on the transition from high to low. This is also called the negative edge, negative-going edge or falling edge of the pulse. To observe this triggering on the negative edge of the waveform, you can connect the console logic probe to the A input of the one-shot and slow down the clock so that as the HI LED on the probe goes out, the pulse out of the one-shot (displayed on the scope) is generated. The Q output of the one-shot will go from low to high and back to low as the trigger trips the one-shot. The $\overline{Q}$ output will go from high to low and back to high as the one-shot is triggered. Each trigger-pulse negative-going edge will generate a single pulse out of the one-shot, and each pulse will be the same width. This width is controlled by the values of resistance and capacitance in the timing portion of the circuit.

The A input is a negative-edge triggered input. The B input is a positive-edge triggered input. Move the console clock output from the A input to the B input. Again observe the Q output of the one-shot. Surprise! No output! Now tie the A input low by connecting the A input to the minus rail of the Superstrip with a jumper. Now the B input should trigger on the rising clock edge. When you use the B inputs of this family of one-shots, the A input must be held low. A positive-going pulse the same width as before (when we triggered the A input) should appear at the Q output, and a negative-going pulse should appear at the $\overline{Q}$ output.

Experiment #37 Varying the Output Pulse Width

Problem: How can the output pulse width be changed?

Solution: Vary the resistance or capacitance.

Procedure: Change the timing resistor (R) of Fig. 1 to a variable resistor. If you have a minature pot, it may be plugged into the Superstrip; if you don't, then use a larger one with wires soldered to the wiper and one of the other terminals and plug the wires into the Superstrip. Observe the effect on the pulse width of the waveform displayed on the scope as the resistance is changed (see Photo 2). The variable resistor is not critical; use whatever you have available.

The value of the capacitance in the timing circuit can also be changed to change the width of the output pulses. Leave the variable resistor at one setting and try substituting different values for the timing capacitor (C) in Fig. 1. Increasing either resistance or capacitance of the timing components will widen the pulses out of the one-shot.

Theory: This particular family of chips is partly linear

Photo 1. Pulse waveform for the Design Console clock.

Photo 2. Dual-trace photograph of the variable-width output pulse in Experiment 37.

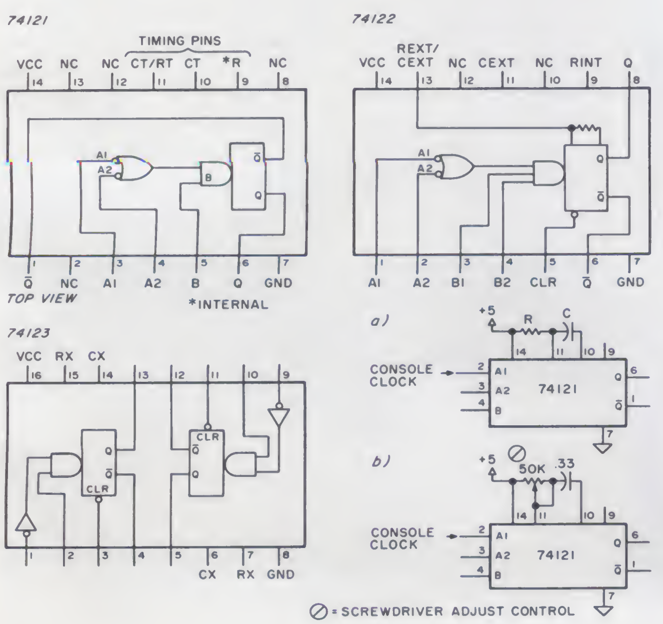


Fig. 1. Circuit diagrams for Experiments 36 and 37.

Photo 3. Experiments 38, 39, and 40 on the breadboard.

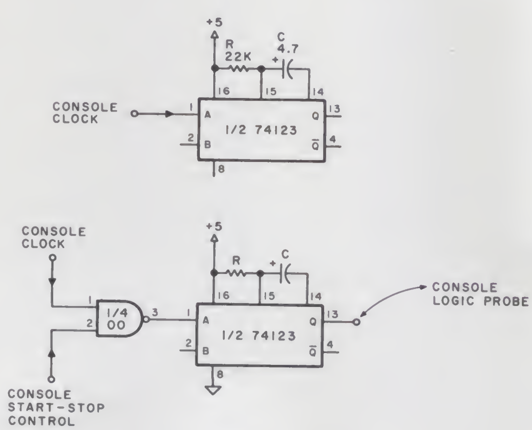


Fig. 2. Circuit diagram for Experiment 38.

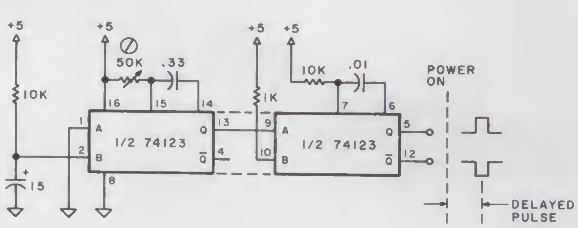


Fig. 3. Circuit diagram for Experiment 39.

and partly digital. They are susceptible to false triggering by noise in a computer. Therefore, engineers do not recommend using them in a computer environment, and suggest that they be "designed around" whenever possible. In spite of this, we find that our computers use these chips whenever pulses must be changed and shaped.

Consider some of the things that can be done with these chips: We can use a rising or a falling edge to trigger them and produce a single pulse out. This output pulse can be tailored to be as wide as we want, or as narrow as we want. (There are limitations of both narrowness and width, but we need not concern ourselves with these restrictions at this time.) Because both outputs, Q and $\overline{Q}$, are available, we can have either a positive or negative pulse out, or we can have both pulses out. In spite of their susceptibility to noise, they have a lot of handy features that make them useful in a computer.

Experiment #38 Retriggerable Operation

Problem: What is the difference between the three chips? What is a retriggerable multivibrator?

Solution: Operate the 74122 or the 74123 in the retriggerable mode.

Procedure: The 74122 or the 74123 may be operated in the retriggerable mode; the 74121 may not be so operated. Increase the capacitance or resistance of the timing circuits enough so that the trigger pulses from the console clock arrive more rapidly than the timing circuit can "time out." When this happens, the Q output will remain high and the $\overline{Q}$ will remain low. The scope should be used with dc coupling to observe this effect. Since some of you will not be using a dc scope, we have designed an alternate circuit that may be used to see the retriggerable operation of the one-shots. Fig. 2 shows the circuit utilizing one half of the 74123 for this purpose.

Theory: Refer to Fig. 2 and Photo 3. R and C, the timing components, require a certain length of time to charge and discharge. If the trigger pulses arrive more rapidly than this time constant (before the circuit times out), the 74123 will trigger again. The net effect is that the Q output of the 74123 goes high on the first trigger pulse, and then will remain high until the pulses are removed. At this time, the Q output will again go low.

The dc scope or the console logic probe is connected to the Q output. The logic probe LED will show pulses as the speed control of the console clock is gradually increased. The LED will appear to be on longer and longer as this speed is increased. Finally, as the trigger pulses arrive faster than the timing circuit times out, the logic

Photo 4. Dual-trace photograph of the "cheap shot" output (Experiment 40).

Photo 5. Dual-trace photo: Upper trace is console clock; lower trace is square-wave output of the 7473.

probe high LED remains on. The 74123 is now being triggered faster than R, and C will time out; and the Q output appears to be high continuously.

The $\overline{Q}$ output will appear to be low continuously at this point. By using the console start-stop control and a 7400 gating circuit we can start and stop the pulses instantly with a push of the switch to see that the Q output is actually being driven high by the trigger pulses. Stopping the pulses with the gate allows the R and C to time out and the Q output to drop low again. Enabling the gate with the start-stop control will again trigger the Q output high, and because the trigger pulses are arriving too rapidly, the Q output will remain high until the triggering stops. This type of circuit is used to debounce keyswitch closures, or to create much wider pulses from a series of very narrow pulses.

The 74121 is a monostable multivibrator and is not retriggerable. Once the 74121 is fired, its inputs are *locked out* and it cannot be retriggered until it has timed out. If the inputs of the 74122 or 74123 are connected to their outputs in an attempt to lock out these inputs, I think that you will find that your one-shot has turned into an oscillator. In fact, if the outputs of a 74122 or a 74123 are connected back to their inputs, you will have another clock generator circuit.

Experiment #39
Pulse Delay

Problem: How can I generate a slightly delayed pulse? For example, how can I generate a reset pulse that will reset my computer shortly after I turn on the machine?

Solution: A one-shot may be used to generate the delay, and a second one-shot generates the reset pulse.

Procedure: Connect the circuit in Fig. 3 on the console breadboard.

Theory: The B input of the first section of the 74123 is used to sense the rising voltage when the power is first turned on. When the voltage rises high enough, the B input will trigger. The first Q output then shows a positive-going pulse whose width (delay) is determined by the R and C of the timing components. We use the negative edge of this pulse to trigger the A input of the second section of the 74123.

The timing components of this section are much smaller and will generate a fast positive pulse at the Q output of this section or a fast negative-going pulse at the $\overline{Q}$ output of the second section. Depending on your particular CPU, one of these two pulses can be used to effect a power-on reset. But, you say that you don't have a dual monostable? Okay, read on. . . .

Experiment #40
The "Cheap Shot"

Problem: How can I generate a pulse without using an expensive monostable chip?

Solution: Use a "cheap shot."

Procedure: See Fig. 4. TTL gate elements are often used to generate pulses. We will use the 7404 for this experiment, but any of the gate elements may be used. Connect the circuit in Fig. 4 on the console breadboard and observe the output on the scope.

Theory: The resistor from pin 3 to ground pulls this input low. Therefore, pin 4 will be high. The positive edge of the signal coming from pin 2 is capacitively coupled to pin 3. As pin 3 is driven high, the output on pin 4 is driven low. The result is a very narrow negative-going pulse virtually independent of the width of the input trigger pulse on pin 1 (see Photo 4). If a positive pulse is desired, the output of pin 4 is run through another section of the 7404 and inverted. Within narrow limits, the output pulse width may be adjusted by changing the size of the capacitor or resistor or both. The circuit is a one-shot pulse generator when triggered, and uses part of a 7404, hence the name "cheap shot."

In the preceding experiment, we used a dual monostable. If you only have a 74121 or a 74122 available, this circuit could follow the monostable and generate the required reset pulse.

Experiment #41
The Timing Diagram

Problem: The hardware manual that came with my computer kit includes a timing diagram. What's a timing diagram?

Solution: We are now ready to generate some waveforms and see how they relate to each other time-wise. Once we get past this hurdle, we should be ready to tackle the computer timing diagrams.

Procedure: Connect the circuit in Fig. 5 on the con-

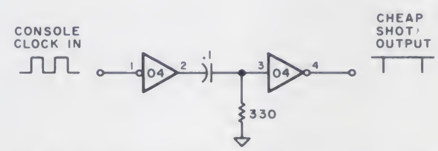


Fig. 4. Circuit diagram for Experiment 40.

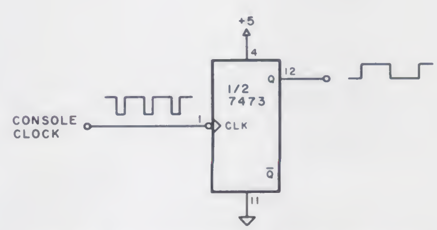


Fig. 5. Circuit diagram for Experiment 41.

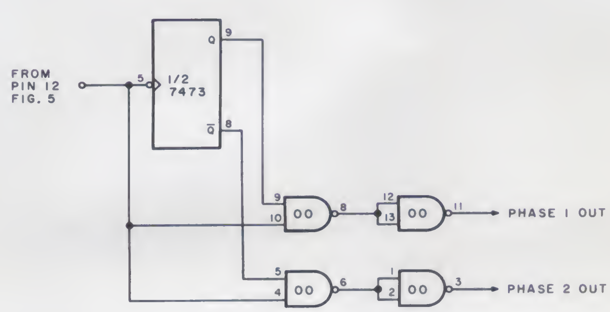


Fig. 6. Additional circuitry for Experiment 41.

Photo 6. Two-phase clock generated in Experiment 41.

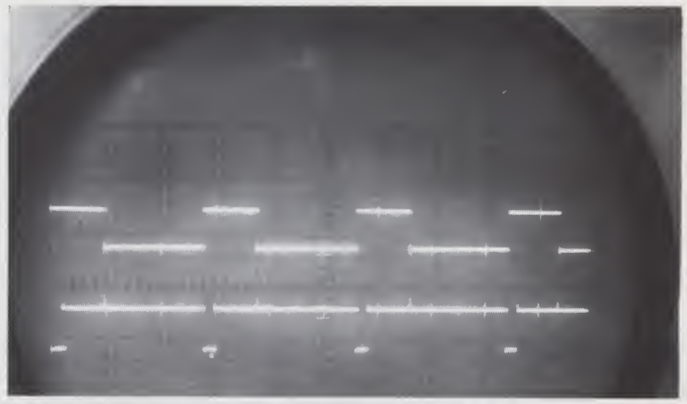


Photo 7. Monostable triggering: Top trace is console clock; lower trace is $\overline{Q}$ output and is triggered by the B input of the monostable.

sole breadboard. The output of the console clock may or may not be a square wave. By running the console clock output through a flip-flop, we can get a square-wave output (see Photo 5). A square wave is characterized by equal on and off times. Observe this square-wave output with the scope.

Next, we add the circuitry of Fig. 6 to that of Fig. 5, generating two sets of square waves that constitute a *two-phase clock*. That is, at any given instant, only one of the pulses will be high; all the other pulses will be low (see Photo 6). The pulses are said to be nonoverlapping. This is the type of clock waveform used to drive the 8080, the 6800 and the 6502 microprocessors.

The two-phase, nonoverlapping clock waveform may be viewed directly on a dual-trace or dual-beam oscilloscope. If only a single-trace scope is available, you will be able to view only one of the clock waveforms at a time.

But wait — there is a way for you to visualize the timing relationships of these waveforms. Lay a sheet of paper over the photo of the square wave. Trace the square wave onto the paper. Now shift the paper up and slightly to one side and draw a second set of square waves. Your reproduction should resemble the dual-trace photo of the two-phase clock waveforms. In this fashion, you can create the timing interrelationship, even without a dual-trace scope.

Fig. 7 shows the timing diagram for the 8080 microprocessor. The top two lines of this diagram show the two clock phases that control the functioning of the microprocessor and, therefore, the operation of the computer built around the 8080 CPU. Each of those positive-going pulses causes the 8080 to take action internally. For example, pulses on the top line (phase one) combine with phase two pulses on the second line and are used to place the address on the address lines and return the data addressed back to the 8080 chip. This constitutes an instruction-fetch cycle for the CPU. The point is that pulses are controlling the internal operation of the CPU (and therefore of the computer system), and are interrelated time-wise. This interrelationship is described using a timing diagram.

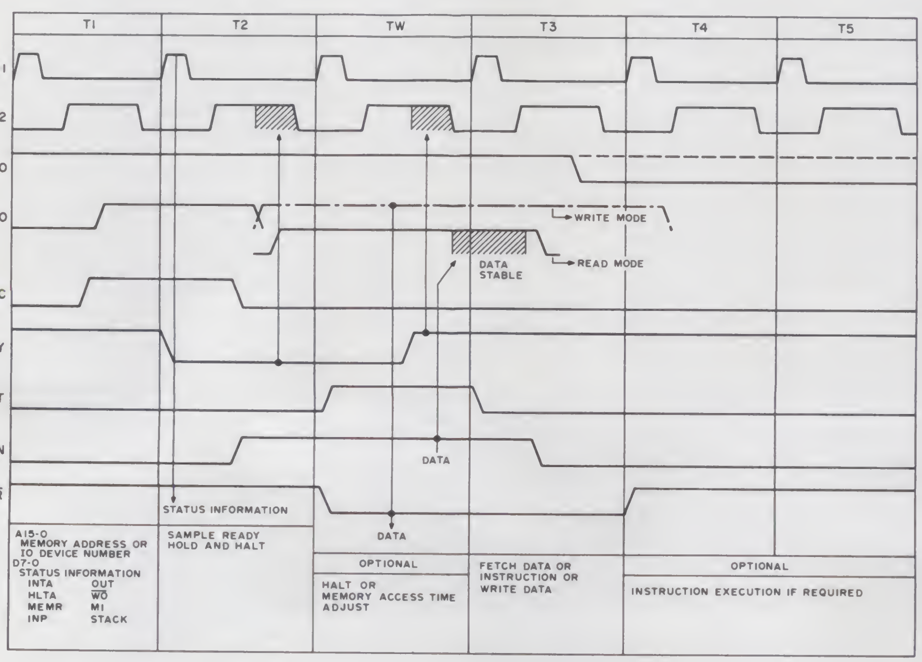


Fig. 7. Timing diagrams for 8080 microprocessor.

Experiment #42
The Schmitt Trigger

Problem: What is a Schmitt trigger and what does it do? What is the funny-looking symbol in the center of the gate symbol?

Solution: Let's get one on the console and experiment with it.

Procedure: Connect the circuit in Fig. 8b on the console breadboard. As the control is varied from one end to the other, the LED will turn on at some particular point. Find this point and then vary the control shaft *slowly* back and forth over this point. It should be possible to dim the LED as you traverse this voltage level with the control. Now replace the chip with the circuit in Fig. 8c. It is the same circuit using a different chip. Again move the control back and forth to turn the LED on and off. This time you should *not* be able to dim the LED. It should be either on or off.

Photo 8. Monostable triggering: Top trace is console clock; lower trace is the Q output and is triggered by the A input.

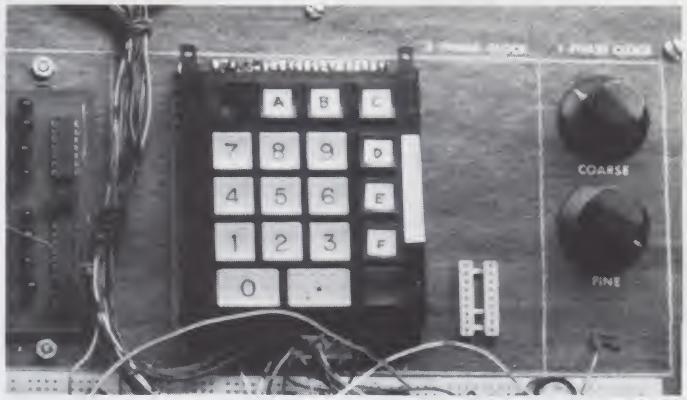


Photo 9. Hex keyboard plugged into Design Console card-edge connectors.

Theory: It is a simple experiment, and any member of the Schmitt trigger family may be used. The hysteresis symbol (inside the gate symbol) comes from the field of ferromagnetics and is used to describe the magnetization of metals. It shows graphically that once a metal is magnetized in one direction, extra energy will be required to effect a field reversal. In semiconductors it means that the turn-on point of a gate is different than the turn-off point.

For Fig. 8b, we can find a point where the 7400 is partially on and partially off. The LED on the output indicates this point by dimming. In Fig. 8c, since the Schmitt trigger has different on/off points, we cannot find any place that will dim the LED; we can only turn it on or off. Why is this important? Why would this kind of device help us in our computer or in any other field of electronics?

Experiment #43
A One-Pulse-Per-Second Generator

Problem: How can a Schmitt trigger be used?

Solution: Let's connect the circuit in Fig. 9 on the breadboard and find a use for the Schmitt trigger.

Procedure: A sine wave input is taken from the console power supply (from the *secondary* of the transformer *— not the primary*). The output should be monitored by the console logic probe.

Theory: A sine wave is not a fast-rising or fast-falling waveform. It gradually rises to a maximum and then gradually falls to zero. It then "rises" negatively to a maximum and again "falls" back to zero. Such a waveform is not a digital logic pulse. A logic pulse has a very fast rise and fall time. If any small fluctuations occur on the rising or falling edges of the sine wave, the TTL logic gates will actually turn on and off several times as the input waveform traverses the critical on/off area.

A Schmitt trigger has different turn-on and turn-off points, and is therefore much more immune to these fluctuations in the input waveform. In fact, a sine wave into a Schmitt trigger comes out as a "perfect" rectangular waveform. Using the scope, compare the waveform at pin 2 of the 74132 with the output on pin 3. The input sine wave is obtained from the console power supply. A couple of my students didn't know where to connect the end of the current-limiting resistor marked 3.3k on Fig. 9, so I added the circuit shown with dotted lines to help you locate the 6.3 V ac sine wave (power transformer secondary). The 3.3k current-limiting resistor was selected so that either an input of 6.3 or 12.6 volts could be used to obtain the sine wave input.

The Schmitt trigger squares up the sine wave so that it will trigger the first 7490 counter. This first 7490 is wired to divide by ten. Its output will be six pulses per second and can be monitored with the console logic probe. The second 7490 is wired to divide by six. The output here, monitored with the console logic probe, will be one pulse every second, and its accuracy will be the same as that of your local power lines: If they are 60 Hz, when divided by 60 with this circuit, the output will be one Hz. This is the way the time base for the clock *chips is* derived, and the circuit in Fig. 9 was the way all electronic clocks were made (with circuits for minutes and hours, of course) before it was cheaper to buy the clock chip with all circuits on it.

Next, I want you to interchange the positions of the divide-by-ten and divide-by-six 7490s. You can do several things to effect this change, and one is very simple. Once exchanged, the circuit will still divide by 60, but it will

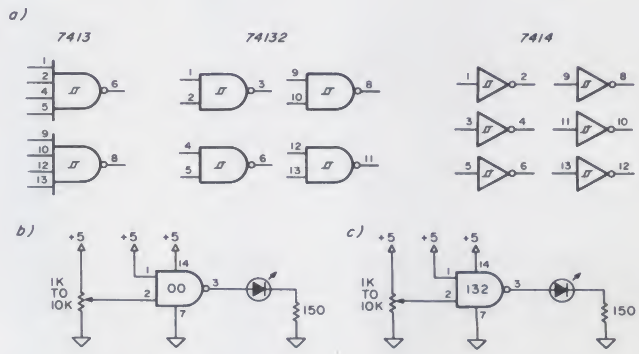


Fig. 8. Circuit diagram for Experiment 42.

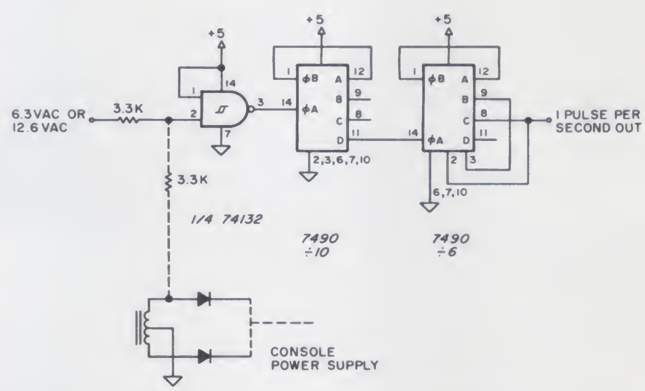


Fig. 9. Circuit diagram for Experiment 43.

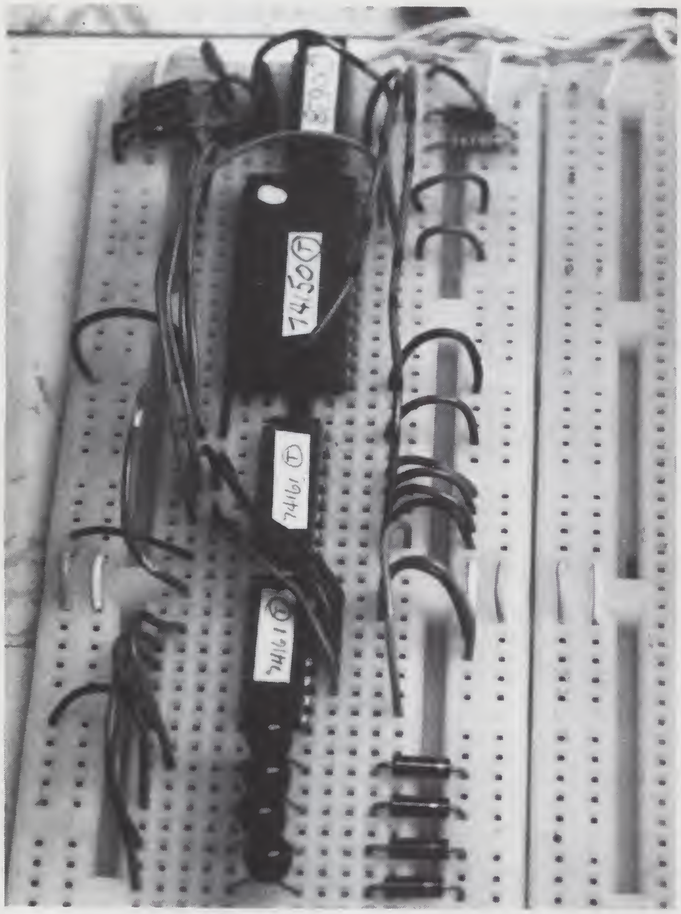


Photo 10. Experiment 44 on the breadboard.

Photo 11. Card-edge connector added to Student Console.

divide by six first and then by ten. This would be its configuration in a clock circuit. Now connect your 7448 decoder to the A, B, C and D outputs, feed the decoder output to the console seven-segment readout, and you will have a ten-second clock. A second 7448 decoder connected to the A, B and C outputs of the first 7490 and a second seven-segment decoder would make a 60-second clock that could be used for a darkroom timer.

Experiment #44
Generating Hexadecimal Code

Problem: I want to build a computer from scratch. I need a method of entering binary code into the computer; I don't have a great deal of money.

Solution: A home-brew hexadecimal keyboard.

Procedure: The circuit of Fig. 10 is to be connected on the console breadboard. This will be the most elaborate circuit that we have yet attempted, with lots of interconnections (and lots of opportunities to make errors) so take your time and proceed carefully.

Theory: We are going to study counters in the next Klassroom, so I will leave the explanation of how this thing works until then. Now, I want to see if you can make the circuit function on the breadboard. The circuit uses two synchronous counters, one as a counter and the other as a storage register. These are marked on the circuit diagram. We are also using a 74150 chip (our first 24-pin IC) as a data selector/multiplexer, which will be explained in more detail later in the series.

I will also give you another clock circuit in this experiment, using the 8T90 IC to provide more drive for the circuit. The 7404 clock circuit can be used here, but one time the 7404 clock circuit did not provide sufficient drive to run this thing, so I had to come up with the 8T90 circuit to make it work reliably.

The circuit will generate the binary code for four binary digits with a single switch closure. Four binary digits taken together can generate a single hexadecimal number. When we interface it to the computer, two successive switch closures will generate a byte of data, and four successive switch closures will generate a computer address (two bytes).

Any single-pole, single-throw switches may be used. There are at least 16 of them, one for each of the 16 symbols of the hexadecimal number system. A good toggle switch costs about $1.50

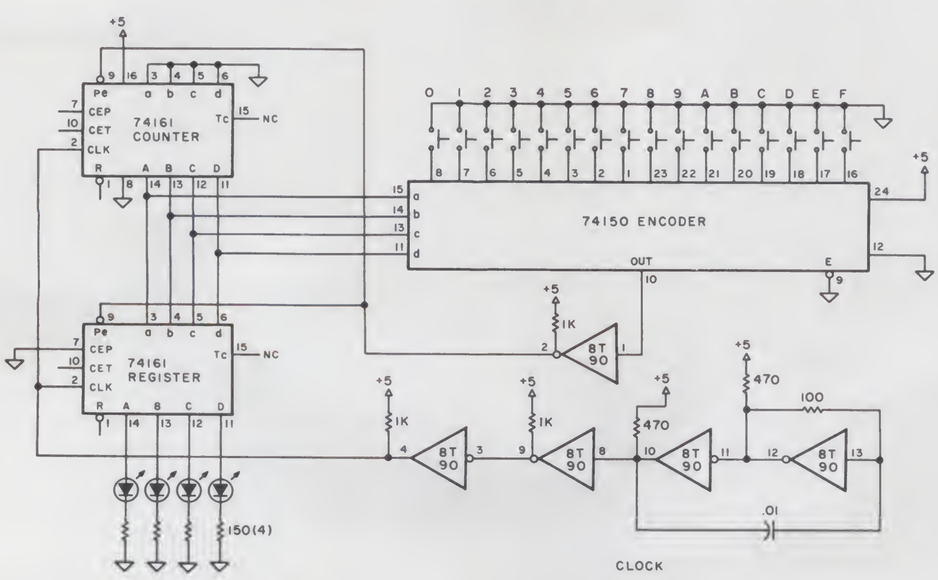


Fig. 10. Circuit diagram for Experiment 44.

to $2 these days, so this will set us back a few dollars. However, the keyboard from a defunct calculator can be recycled for this operation, saving us quite a bit. Even if you have to buy a surplus calculator keyboard, it will cost far less than the toggle switches. Photo 9 shows a salvaged calculator keyboard plugged into the Design Console in two of the card-edge connectors. A circuit board, attached to the back of the keyboard, holds the clock, the 74161 counters and the 74150 multiplexer.

The experimental circuit on the console breadboard may be tested without using switches for the key closures. A wire connected from the 74150 switch pin to ground will generate the hex code, and a bank of four LEDs connected as shown in Fig. 10 will serve as a monitor. Photo 10 shows the circuit built up on the Student or Design Console.

The photograph of the home-brew hexadecimal keyboard shows how it can be added to the Design Console as an auxiliary support function. It can also be added to the Student Console by adding a card-edge connector to the upper horizontal surface of the console (Photo 11). The bolt carrying the power into the console must be moved to the right to accomplish this, but there is plenty of room. This added edge connector will help keep your console from becoming obsolete by allowing you to plug in new functions as needed. The original circuit was published in Fairchild's 1973 *TTL Applications Handbook*.

Preview

In the next session, we will get to registers and counters. You will need the 7490 (which you already have) and at least one member of the synchronous counter family — a 74161, 74163, 74191 or 74193 will suffice. We will also investigate the storage register/counter group, and the 74177 (8281) will be used experimentally. We will also use the 7474 and the 74175 registers experimentally. Sierra Electronics, Box 11, Auberry CA 93602, will offer the following package for Klassroom No. 9: 74161, either the 74191 or 74193, the 8281, the 7474 and the 74175 as a kit for $5 postpaid. Sierra Electronics will also offer the active component parts kit for the hexadecimal keyboard for $10 postpaid. Californians, please remember your six percent state sales tax. ■

John Leslie
10 Souhegan St.
Milford NH 03055

Help for Beginning Programmers

nine basic principles ease the frustration

In the August issue, Wayne Green challenged any old-timers to stop laughing at computer hobbyists who make the same mistakes as big-machine programmers did 20 years ago — at least long enough to write a helpful article.

Now, I really don't entirely fit that category. In the first place, I have not been laughing at computer hobbyists. And second, I may not be enough of an old-timer. But I have been around long enough to make a lot of my own mistakes. I was playing space-war in 1963 and started programming in earnest in 1965. Since then, I have barely stopped long enough to catch my breath.

Principle No. 1: *The best study of programming is programming.*

In the past 12 years or so, I have been asked approximately 7812 times how one gets started in programming. Each time, I have responded that you start by writing a program. (This, for me, is a marvel of consistency. I cannot offhand think of any other question I have answered the same way every time.)

If you run across someone who ridicules you for making mistakes, feel sorry for him. It is quite impossible to become a functional programmer without making at least 5000 mistakes. If you are one of those people who finds his identity shaken by each mistake he makes you had better find another hobby, pronto! Of the hundreds of people who have asked me how to break into programming (perhaps I should explain that I was a physics major; to a physicist, 100 is approximately equal to 7812), I would be surprised to find more than two or three who have become successful programmers. Ninety percent of them, I am sure, have yet to write their first program. Now, I probably shouldn't criticize here. It was well over a year after I was introduced to computers before I wrote my first program. The thing I remember that stopped me was a feeling that it would be a terrible waste of expensive computer time to run any program I might write. Recall that in those days, computer time cost over $100 per hour.

Well, none of you reading this have that excuse. Many of you have your own computer, or know someone else who does. Computer time is as close as you can get to free.

The next thing I remember holding me back was that I couldn't think of anything I wanted the computer to do. This problem is certainly still with us. I suppose I could give some examples of first computer problems here, but I'm not going to. What you do for your first program is of utterly no importance. Personally, I haven't the remotest idea what my first computer program did. Just don't choose anything particularly important. After all, it isn't going to work.

Principle No. 2: *"Programming is like pancakes: throw away the first one."*

In the past year, I have read at least 20 articles saying (among other things) that you should plan out all details of a computer program before writing the first line of code. This occasionally may be good advice; I must admit that it follows my actual practice on many occasions. But for the beginner, it is the worst possible advice.

The novice programmer, you see, has even less idea how to design a program than he has how to write it. (Granted, we may be dealing in infinitesimally small quantities here.) To find out whether the program works, he just runs it. To find out whether the design is good, he would have to talk to an experienced programmer and probably wouldn't even be able to understand the language. So don't waste your time worrying about the design. Dive in and write a program. Once it is written, try to make it run. And once you can make it run (more or less), throw it away. Yes, I know that hurts — all that time going to waste. But wait! Think for a minute. What was the purpose of this exercise?

Give up? The purpose was to learn about programming. And if you actually managed to get a program running, you have learned a lot. So, the exercise was a success. You have not been wasting your time.

Now, if it turns out you made a poor choice for a first program, and in fact you could actually make use of such a program, still throw it out. But take advantage of this opportunity to learn a little bit about design. One thing you will have learned is the actual nature of the task you set out to make the computer do. When you started out, you no doubt thought you knew just what you wanted. If you go back over what decisions you had to make, though, you will find how pitifully little you actually knew of the task when you started. But now you know. And since you know, you have some basis for designing a program. So before you write the second version, spend a little time breaking the task into pieces. Draw a flowchart if you want, but keep it all on one page.

Principle No. 3: *It's always harder than it looks.*

This is not what you want to hear. You want to be reassured that all the trouble you have been having will go away with time. Well, I can

reassure you there: it will. But new problems will crop up. The next time you hear a programmer saying some task was straightforward, check to find out what he means. The only thing that conceivably could be written without having to back up at some point and change the way you did something is a task you already fully understand. And those are rare.

If you have the right outlook, though, there is no reason to despair just because it is harder than you thought it would be. That just means you have to slow down and think it over. And it always shows that you now understand the task better than you did when you thought it was easy.

Analyze a bit to find out why it seems harder now. Many times it is because you have found a flaw in the method you were using. If so, immediately look for other flaws in that method. One of those 20-year-old mistakes is the "special case." Twenty years ago, whenever programmers found a flaw in their methods, they assumed it was just a "special case" and put in code to check for that special case and to do a different computation for it. The result of this was programs that computed a supposedly continuous function with half a dozen (or more!) different computations, based on the input value. This can lead to disaster many ways. For instance:

1. Programmers would sometimes neglect to test that case again after the fix or would test it incorrectly. Result: a time bomb, giving the wrong value the next time that input appeared.
2. Usage of that function might take the difference of closely spaced values, greatly magnifying the discontinuity. Result: inaccurate results, miserably difficult to trace.
3. In transferal of the program, a command might get garbled, affecting only that special case. Result: another time bomb, sprung on a user who has no understanding of the program at fault.

Programmers did this (creating special cases) because they were in a hurry and because the task was supposed to be easy. You have no excuse to be in too much of a hurry (you have no boss breathing down your neck); and you know that the task isn't as easy as it once looked. So don't try to fix things by creating "special case" code.

Programmers also used to create special cases when they thought they could save a little computer time by doing so. They were almost always wrong, if you count the computer time spent in debugging. It is hoped you won't feel too tempted to do that because each special case eats up memory space, which is usually in short supply for the hobbyist.

Principle No. 4: *Don't waste your time trying to make the computer run faster.*

I must immediately cite an exception. Sometimes the execution time is simply unacceptable, and you must do something to fix it. But this is quite rare; even then it should be fixed after the program works.

Computers 20 years ago were slower than today's microcomputers, so programmers must be excused for all the time they spent studying tables of execution times. I have finally cured myself of this habit (subject, of course, to occasionally working out timing on paper for a microcomputer I don't have access to). But one still sees articles comparing the relative worth of several microcomputers on the basis of their execution times. Such articles are valuable, but the novice should firmly understand that speed is just one of many factors and is hardly ever the most important.

It is far worse when the novice starts to think he should take advantage of the full speed capability of his microcomputer. There are a lot of tricks you can learn, and it is indeed fun sometimes to try to pry loose another microsecond. But when you are writing a computer program to do something, keep in mind that the most important thing is to get it done.

Twenty years ago, it was common practice to store a value away in some obscure, supposedly temporary, register, and use it in code several pages away from where it was stored. This, after all, saved perhaps 20 microseconds. But all too frequently two to four hours of computer time were wasted trying to hunt down the bug.

We old-timers see a lot of this in today's computer hobbyists. Sure, it's a nice trick to save something away in the H and L registers of the 8080. But if you fail to comment what you've done, somebody's likely to get mighty confused about the way the program works. It's better to leave out such tricks unless you really are cramped for time.

Most successful programmers have developed the habit of noting, either explicitly or implicitly, which temporary registers are used by each subroutine. A common general rule is that each subroutine is expected to save and restore any temporary registers it uses. This

means that at the beginning of the subroutine each temporary register is stored in some temporary area, and before returning, those saved values are loaded back into the registers. This method uses a lot of time and memory, but can save your sanity, surely more precious.

Alternatively, you could accept the convention that each subroutine may clobber any and all temporary registers, and that the calling code should do any necessary saves and restores. This takes less time and sometimes less memory, but is error prone, in that the programmer is too apt to remember that a particular subroutine doesn't clobber some register and omit the save and restore.

Principle No. 5: *Don't cut corners.*

When writing a program, it is exceptionally difficult to avoid thinking that some operation doesn't need to be done "in this case." Remind yourself that to omit it is to create a special case. This is perhaps the strongest area of temptation, since it requires more of your time, more execution time and more program memory to include the operation. But it will pay off handsomely if you make the rule that every operation is to be done unless some pressing need prevents it, and that even then a clear notation is required. If you are testing for arithmetic overflow, test after every arithmetic instruction, not just after those you expect might overflow. If you are calling a subroutine that is entitled to clobber something, include code to save and restore, even though you know the subroutine, in fact, doesn't clobber it.

Now, probably I am just about to lose a whole lot of readers, who are saying, "How does this nut expect me to do all this within my 4K BASIC?" Frankly, I don't. If you want to be serious about programming (much of anything beyond copying someone else's BASIC program), you are going to need a lot more memory, or you are going to need to spend enough time to learn how to do things in machine language. Something that will fill 4K BASIC can most likely be done in 1K in machine language. And you don't need the BASIC interpreter in memory to run machine language.

I have no particular desire to force machine language down anyone's throat. For typical hobbyist applications, it is often a rather poor choice. But, even if you obey my advice literally about cutting no corners, you can fit many more program features into available space. If you can afford enough memory to run complicated BASIC programs, by all means get it. If not, I recommend machine code. It does not require any great intelligence to learn to use it — just perseverance.

There is, of course, a middle ground: a high-level language compiler. In principle a compiler can generate reasonably compact code, and vacate its own memory space during execution of the program, giving you most of the advantages of machine code. In practice, there are problems of implementation, which make it difficult to write a compiler for a hobbyist system. Now, the choice is bleak. I hope this situation will improve substantially within a year or so.

Principle No. 6: *Fix the right problem.*

Programmers 20 years ago regarded the memory size of their computer as sacred. If their programs caused the compiler to strangle with too many symbols, they dutifully rewrote the program with shorter symbols or doubled up on variables. As you can imagine, this made for many quite unreadable programs.

The first assembler (machine-language compiler, more or less) that I ever worked with had an ingenious solution. All its symbols were a single letter followed by an integer, and the user had to declare the size of the symbol table beforehand. Presto! no symbol table overflow was possible. And it ran fast!

But if you can imagine a 10,000 line program with no mnemonic labels (was that Q231 or Q132?) you quickly see that it becomes unreasonably hard to read. So a local programmer wrote another version, which accepted six-character labels, and soon that assembler (perhaps 1/10th as fast-running) became the one most used. In due course people began to realize that six-character labels were a little too restrictive in larger programs; but as far as I know, that was not changed before the whole computer was replaced.

We are seeing much the same thing with hobbyists. If your BASIC program doesn't fit in the available memory, we see you dutifully stripping off all the comments — which somehow reminds me of a program called CMPRS (system names were restricted to five characters). The purpose of this was to take an ordinary ALGOL program and compress it into the fewest possible characters. I'm not absolutely certain, but I think the main reason had to do with disk allocation: the program took up less disk space when compressed. As you can imagine, since ALGOL had no restrictions about the number of statements on a line, the result was nearly unreadable to a human.

But this horror story has a happy ending. Someone realized the difficulty and set about writing a program to reverse the process. It was called XPAND, of course, and it carefully indented lines to show the actual block structure and loop structure of the program. So the net result of running the two of these was often a more readable program than the original. Nowadays, large-scale computers always come with enough memory to support a ridiculously large program. After all, memory is cheap, and they come with enough mass storage (disk, etc.) so that people feel no desire to compact their programs into unreadable form.

All of this simply means that if you are running out of space, you need more memory — not more obscure programs. And if you are running out of time, you need a faster computer or a different method. You may have to get by for a while with the hardware you have, but never forget what the problem really is. Because that is what you are going to have to fix, sooner or later.

Principle No. 7: *"We have met the enemy, and he is us."*

Almost all novice programmers seem to assume that the enemy is the computer when they set out to write a program. This simply is not true. The computer is going to do exactly what it is told; no more and no less. If it had anything resembling feelings, it would desire to please you.

Alas, the enemy is yourself. Your brain is a marvel utterly beyond comprehension, but it doesn't think the way a computer works. It is capable of the same processes, but it has long since discarded them as ridiculously slow and inefficient. Someday, we will better understand the processes by which the brain does operate. And we will set out to build computers that work that way. In the meantime, whenever you want to program a computer, you must think of a way to do the job using only the things the computer knows how to do.

The situation is complicated because what the computer can do, it can do many times faster than the human brain. This means that the practical solution tends to involve a combination of operations that the brain is virtually incapable of doing itself. The best programmers learn to model these opera-

tions in their heads and work with the models. If you find yourself learning to do that, you might consider a career as a programmer. If not, I suggest you keep your current job. But don't give up computers. You can be a successful hobbyist without that skill.

Principle No. 8: *If at first you don't suceed, try a smaller task.*

Various studies have shown that there is an abrupt drop in comprehension whenever a concept is presented on more than one page. If you have been writing a program or designing one, and find yourself bogged down after you have written a few pages, it is definitely time to break the task down into pieces. The (as close as can be expected to) perfect tool for this job is the subroutine.

Subroutines were called into use over 15 years ago in order to save space (actually, many useful things came from our long avoidance of solving the memory problem with more memory); they quickly became popular. Gradually, we have learned to make effective use of them. And gradually, the hardware implementation is becoming better (for instance, most subroutine calls now save the program counter in the stack). But too many people still think of the subroutine as just a tool to save memory. And they follow that to the logical conclusion that they shouldn't write something as a subroutine until they know they will call it at least twice. Consequently, they never take advantage of its more useful role.

The true advantage of a subroutine is that it needs absolutely no knowledge of the design or implementation of the routine calling it. All the information it needs is contained in a very few temporary or interface registers. Conversely, the calling routine needs absolutely no knowledge of the design or implementation of the subroutine. It merely must set up parameters and read results. It is, of course, possible to accomplish this result without using the hardware subroutine call instructions. And the result would be just as desirable. That's not really the point. It is simply easier to do it with the subroutine call instructions.

So, take your bogged-down program, and start carving out some subroutines. Any logically distinct section that seems to require the better part of a page of code is a good candidate to be turned into a subroutine. It will take a lot of practice on many different programs before you learn to carve subroutines optimally, but keep at it. When you become proficient at carving subroutines, your whole program will be a lot of subroutines, each on its own page, called in strict hierarchy up through the main control program, also contained on one page. But don't worry about that now. Just break off enough pieces so that you can comprehend what is left and start work on what is yet to be done.

Principle No. 9: *Enough is enough.*

When you get too tired, stop and rest. When you come back to it tomorrow, or whenever, you will find you can't remember what you were doing. Don't worry. Put the necessary time in and you will come to understand it again. You will also unearth some of the errors you were making because you were too tired.

Speaking of too tired, I think this article has run long enough. Having done my duty, I will go back to laughing at programmers making 20-year-old mistakes. What? You thought I wasn't laughing at computer hobbyists? But I wasn't! I was laughing at all the professional programmers who are still making 20-year-old mistakes. ■

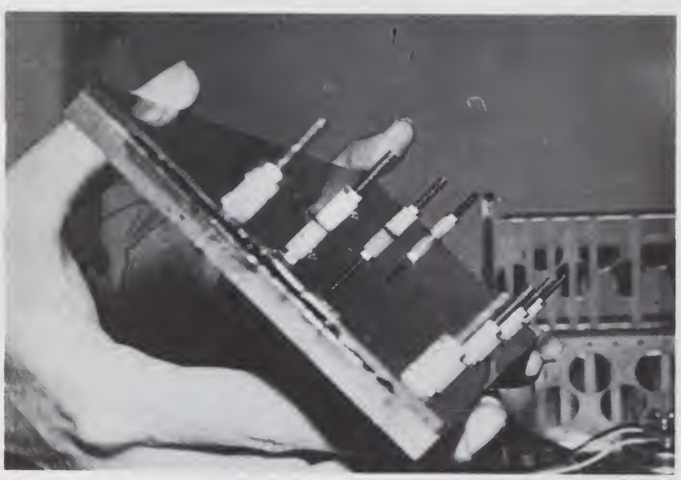
J. Ward Spaniol
11940 Laurelwood Drive
Studio City CA 91604

Your Imsai and You

four steps to a happier relationship

Over the last 18 months, I have been mildly irritated by some peculiarities of my Imsai 8080. They are somewhat trivial, and all have to do with the front panel. In no particular order, they are:

1. When disassembled, the front panel has a tendency to scatter nylon spacers all over kingdom come. The simple solution is to leave the thing alone.
2. The machine locks up when you blunder into an unanticipated halt instruction. This can be remedied by turning power off and on at the expense of severe solid-state amnesia accompanied by occasional temper tantrums. It can also be exited by a simultaneous stop and reset command.
3. The single-step function performs one machine cycle instead of the expected single instruction. This makes it possible to step part way through an instruction, examine some other location and run . . . ad infinitum. However, this situation can be avoided if you religiously reset after single-stepping.
4. The programmed outport lights in the right-hand corner respond by being off when a logic 1 is written to them. It offends my sensibilities to have to remember that a dark display is all ones, or the power is off.



1. Your front-panel sandwich assembly should look like this when you're preparing to glue it together.

There are, of course, other annoyances, but since I don't know how to remedy them, they will be left as a student exercise. The four gremlins identified, however, can be dispelled in an evening. About the only thing you may not have handy is a good all-purpose cement. I used Weldit. If you should happen to park overnight on a tube of it, you might destroy your car's tire by cementing it to the driveway. There may be other equally good cements, but I haven't run over any of them.

The Disassembly Problem

Begin by disassembling the front panel. I have a Molex connector to bring in the ac power, which saves soldering and unsoldering. If you don't have something similar, I suggest you wrap the power leads together and insulate them, using the line cord as a power switch during this operation. In any case, always pull the plug before pulling the front panel. Count your hardware to be sure you have eight sets each of: long spacers, short spacers, screws, nuts. You'll also have separated the front-panel sandwich from the front-panel board.

Now, reassemble the sandwich and the long spacers with the screws and nuts. Make sure the nuts are fairly tight and you use your cement liberally around the spacers and the red plastic. Be very careful not to get cement on the screws and nuts. Add a small amount of cement on the short edges of the sandwich. If you ever want to separate the sandwich after this has been done, just slice the edges with an x-acto knife. See Photo 1 for this assembly.

When your sandwich dries (about an hour) the four-layer stack and the eight spacers will be a solid unit.

The short spacers are cemented to the back of the front-panel printed circuit card, using the same technique with the mounting screws. The cement is non-

conductive, so you needn't worry about the PC traces. Again, I urge caution about stray cement on the screws and nuts.

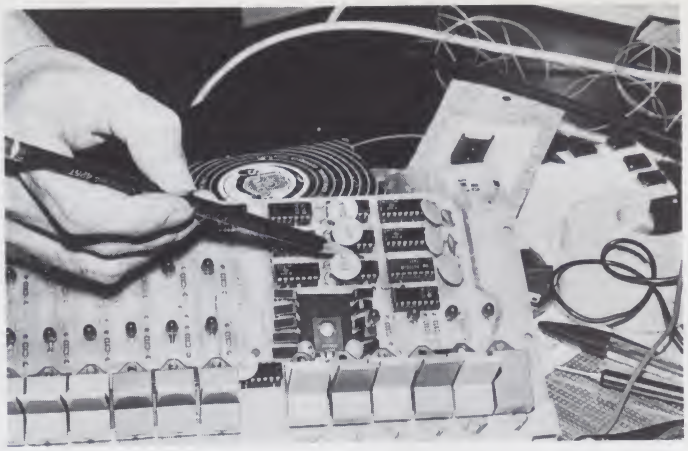
Eliminating the Halt Problem

Next, refer to the run/stop portion of the front-panel schematic. Note that flip-flop U22 is reset by a complex function of the stop switch appearing at pin 10 of that flip-flop. This signal occurs only at the beginning of an instruction when the stop switch appearing at pin 10 switch is pushed. There are no other instruction beginnings (fetch) after a halt instruction, so the stop switch cannot get the machine out of the run mode. The combined stop and reset will cause the machine to go to memory location zero and begin an instruction. This then gets it out of the locked-up state. I used a 1N4000 diode to bring the reset switch contact into pin 10 of U22 (see Photo 2). A reset from the front panel will reset the run flip-flop and have the same effect as the combined stop and reset. I mounted the diodes close to the reset switch, covered it with nylon tubing and cemented it in place. The cathode is soldered to the lower pad of the switch, and a wire is run from the anode to U22 pin 10.

At this point, the panel should be tested. I recommend a card extender placed in the first card slot rather than putting the front panel in its usual position (see Photo 3). I don't have nerve enough to use the extender card in the front-panel slot since it doesn't have supports and could easily lean back into the sheet metal behind it. Some front-panel functions may not work with the front panel on the extender . . . do not panic as this is only caused by the extender changing the bus characteristics slightly. Exercise the problem (and cure) loading a halt, hex 76, at location zero, and then execute. When the machine locks up the reset switch alone should restore everything.

A True Single Step

The next change will require a little surgery on the face of the printed circuit board (the side with the components on it). Using a scalpel or an x-acto knife, carefully cut the trace coming down from U19 pin 13. Verify this by testing for an open circuit from U19 pin 13 to U22 pin 13. Now add the wiring on the back of the



2. Combining the stop and reset functions requires addition of a diode.

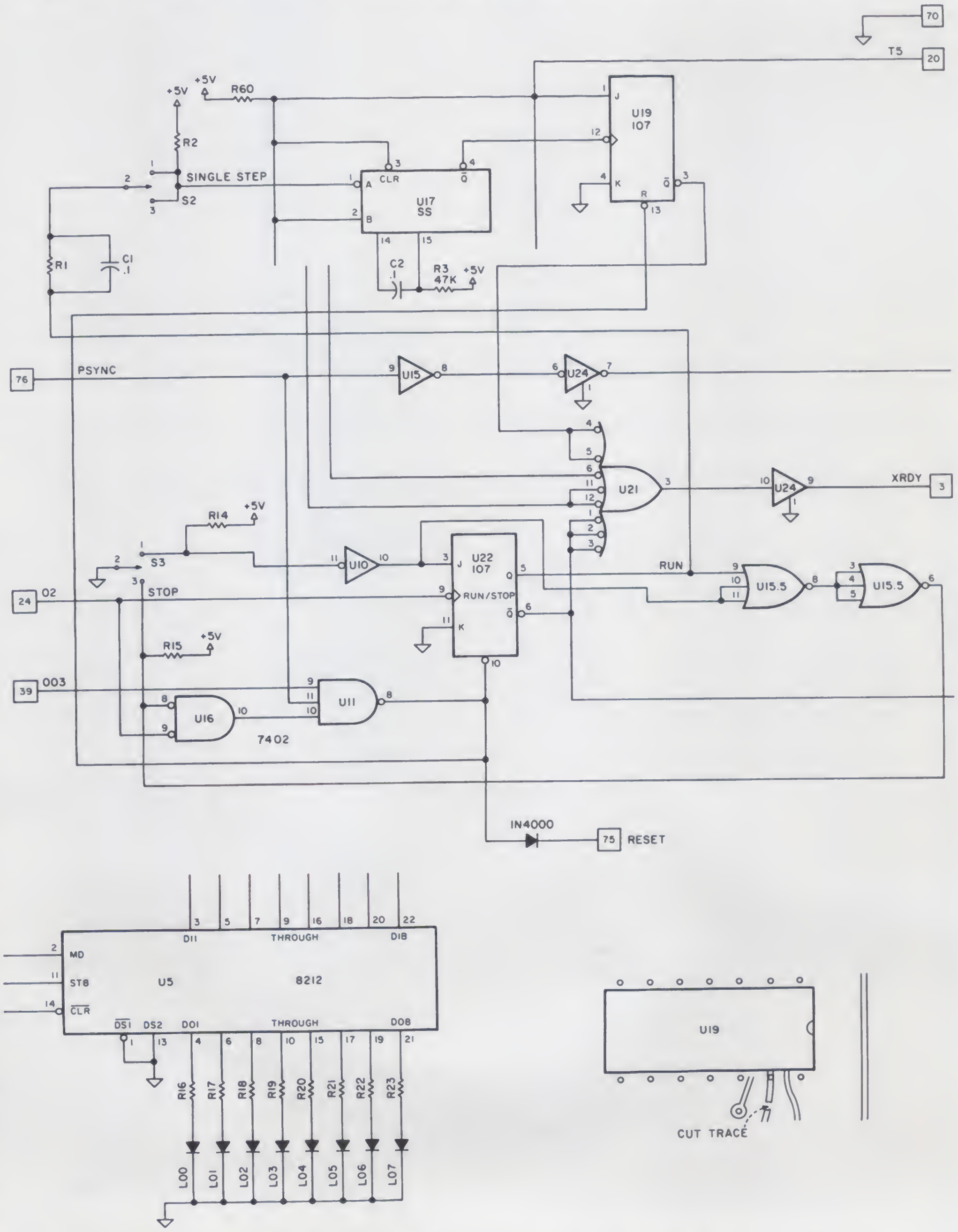


Fig. 1. After alterations are made to the Imsai front panel, mark your schematic accordingly.

3. Using an extender card to test your modification is the safest method.

board to complete the circuitry as shown in Fig. 1. Use 24- or 26-gauge wire and, after soldering, spot-cement them to the board.

This change uses two spare gates in U15.5 to develop a reset signal to U19 pin 13 at the beginning of a new instruction, rather than the start of a machine cycle. The net effect is to do a complete instruction for each cycle of the single-step switch. The verification program I used is:

Location 0 Jump to 4
Location 4 Jump to 0

Previously it would have required six single steps to go around this loop once. Now it will only take two. The databus display will show a constant C3 and the address display will alternate between 0 and 4.

Programmed Output LEDs

The final alteration is the most tedious — reversing all eight of the programmed output LEDs. Unsolder and remove them one at a time, clean out the holes and solder them back in with the leads reversed. The flat spots on the LEDs should now point toward the top of the board instead of the bottom. The common +5 volt bus is isolated by cutting the traces at the left and right side of the display. A small grinding tool is handy for this since these are fairly wide traces. Restore the +5 volt continuity by wiring around the now-isolated display. (I ran a wire directly under the status display LEDs up to the bottom lead of the capacitor between U10 and U11, connecting from the +5 volt trace on the left side of the board.)

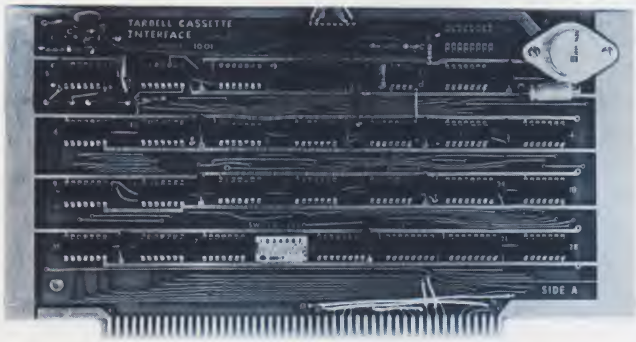
Last, connect the isolated bus to logic ground at any convenient point. I used a resistor lead on the back of the board to bridge the bus over to the ground trace in the center of U10. The LEDs will now light when a logic one is present, instead of a logic 0. The program I used to test this change simply increments the accumulator, outputs to FF and loops. The display output should now count up instead of down. Before trying this, inspect the LEDs carefully, since any installed backward will be lost and gone forever. You can be extra safe by marking the top of each one with a cement dot before you start reversing them. The mistake I made was to reverse the same LED twice.

Summary

Before you relax in triumph, red-line your schematics to reflect the modifications. This bothersome trifle seems insignificant with a personal computer, but considering that this machine has a potentially infinite life span, accurate documentation is a necessity. Who knows . . . maybe the computer log book will pass from generation to generation, like the family Bible. ■

TARBELL SETS STANDARDS
For Hobbyists and Systems Developers

Sales to thousands of hobbyists over the past two years have proven the Tarbell Cassette Interface to be a microcomputer industry standard. Tarbell Electronics continues research and development to produce new and efficient components to fill hobbyists' changing needs.



TARBELL CASSETTE INTERFACE

- Plugs directly into your IMSAI or ALTAIR*
- Fastest transfer rate: 187 (standard) to 540 bytes/second
- Extremely Reliable—Phase encoded (self-clocking)
- 4 Extra Status Lines, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable of Generating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4

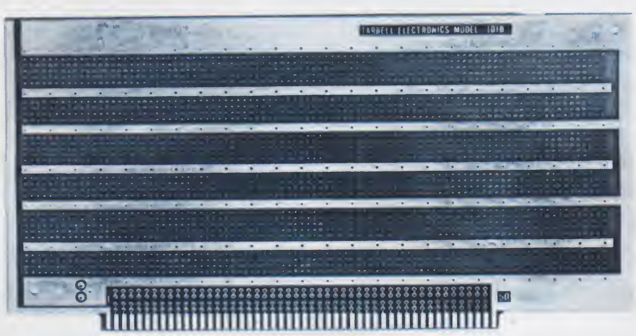
TARBELL FLOPPY DISC INTERFACE

- Plugs directly into your IMSAI or ALTAIR* and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K bits per second on normal disc format capacity of 243K bytes.
- Works with modified CP/M Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI Chip.
- 6-month warranty and extensive documentation.
- PRICE:
  Kit $190 . . . . . . Assembled $265

Compatible Disc Drives

Ask about our disc drives priced as low as $525.

TARBELL PROTOTYPE BOARD
Model 1010



- Gold plated edge pins
- Takes 33 14-pin ICs or
- Mix 40-pin, 18-pin, 16-pin and 14-pin ICs
- Location for 5 volt regulator
- Suitable for solder and wire wrap
- ALTAIR/IMSAI compatible

Price: $28.00

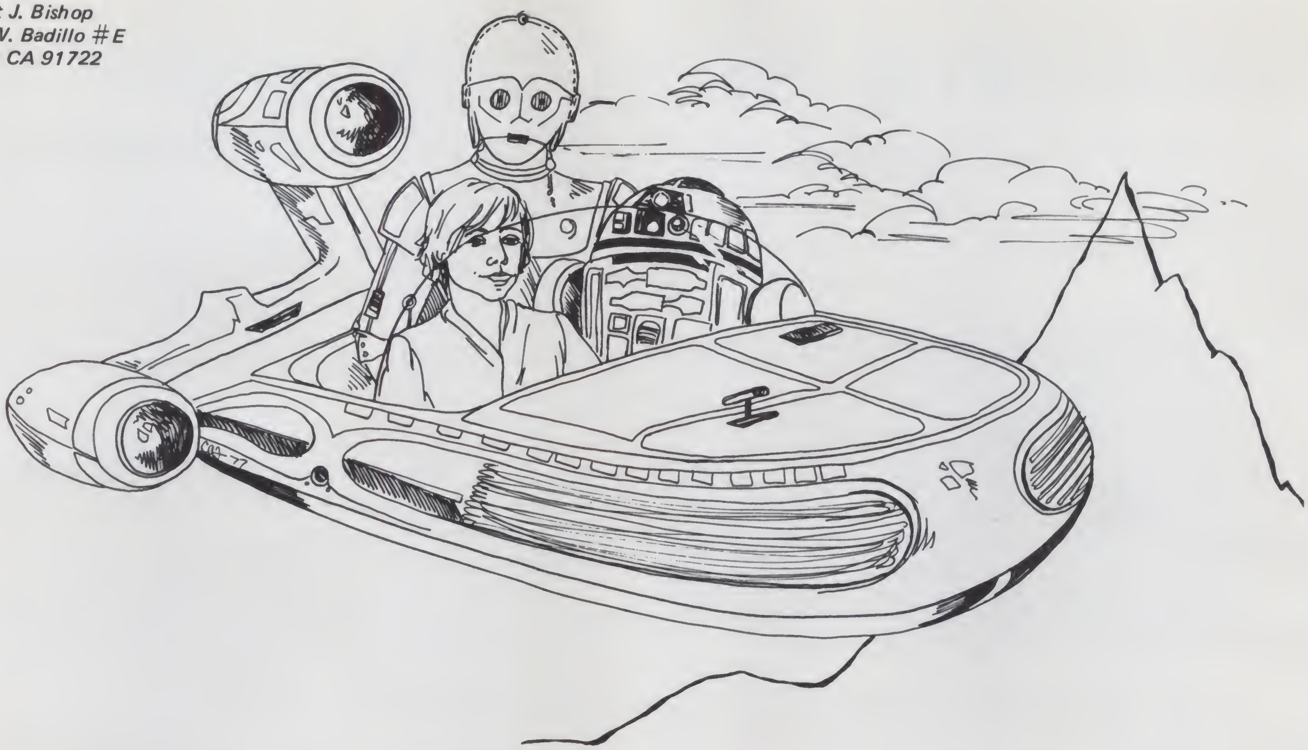
For fast, off the shelf delivery, all Tarbell Electronics products may be purchased from computer store dealers across the country. Or write Tarbell Electronics direct for complete information.

*ALTAIR is a trademark/tradename of MITS, Inc.

Tarbell Electronics
20620 South Leapwood Avenue, Suite P
Carson, California 90746
(213) 538-4251

T11

Robert J. Bishop
1143 W. Badillo #E
Covina CA 91722



Rocket Pilot

an interactive game for the Apple-II

Program listing.

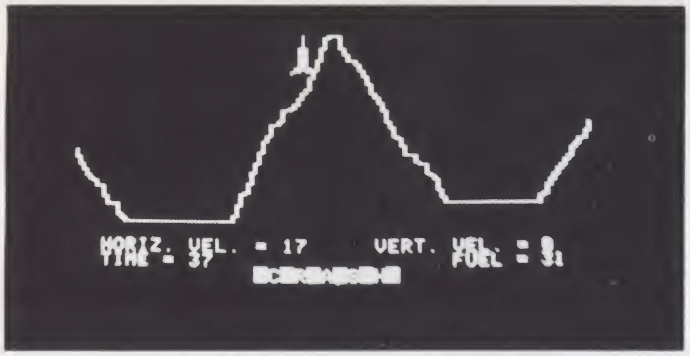
```
  0 REM *** ROCKET PILOT ***
  1 REM COPYRIGHT 1977, R. J. BISHOP
  5 N=7
  7 XMAX=255: YMAX=255
 10 GOSUB 5000
 20 CALL 2048+169
 25 TIME=0: FUEL=250*N
 30 GOSUB 1000
 33 PRINT "";: REM "BELL"
 35 GR : POKE -16297,0
 40 YL=100*YL: YR=100*YR
 50 X=2400: Y=YL-2000
 60 DX=0: DY=0
 65 DDX=0: DDY=N
 70 POKE 18,0
 75 CALL -936
 80 VTAB 22: PRINT "  HORIZ. VEL. =         VERT. VEL. ="
 85 VTAB 23: PRINT "  TIME =";: TAB 30: PRINT "FUEL ="
 90 GOTO 140
100 DDX=(2*N* PDL (0))/XMAX-N
103 FOR K=1 TO 10: NEXT K
105 DDY=(2*N* PDL (1))/YMAX-N
110 DX=DX+DDX
115 X=X+DX: IF X<0 THEN X=0
120 DY=DY+DDY
130 Y=Y+DY: IF Y<0 THEN Y=0
135 CALL 2048+260
140 POKE 16,X/100: POKE 17,Y/100: POKE 19,0: CALL 2048+256
145 TIME=TIME+1: FUEL=FUEL- ABS (DDX)- ABS (DDY-N)
150 IF FUEL<0 THEN FUEL=0
155 GOSUB 4000
160 IF X<=0 OR X>=13300 OR Y<=0 THEN 300
165 IF PEEK (19) THEN 900
170 IF X>7000 AND Y>YR-2000 THEN 800
175 IF X>7000 OR Y<YL-2000 THEN 400
180 X=X-DX: DX=0
185 Y=YL-2000
```

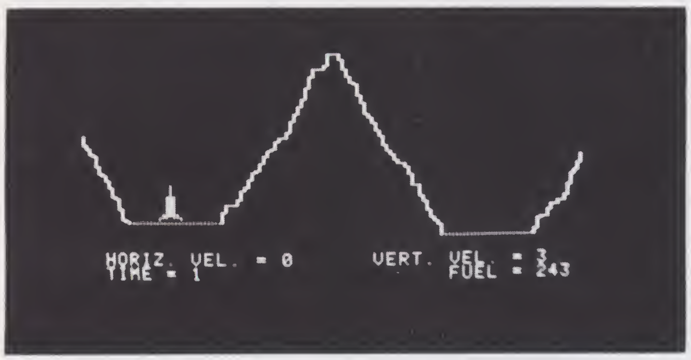
There are many Lunar Lander-type games available for microcomputers these days, and most of them have similar rules of play: You type in some numbers, and the computer types some other numbers back at you. (Kind of sounds like *most* computer games, doesn't it?) Furthermore, most of these games are one-dimensional (only up-and-down motion) and non-real-time (you can sit and think for as long as you want before making your response).

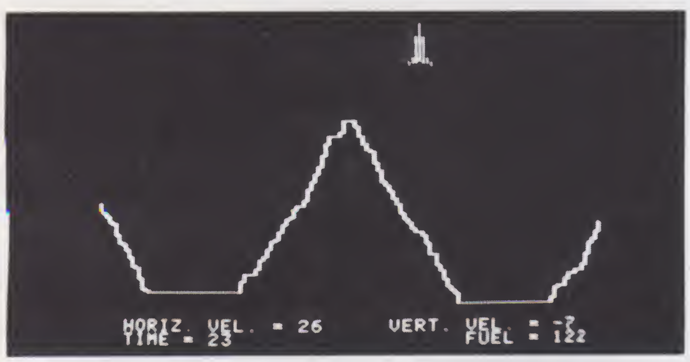
Tired of these "type-something-in/type-something-out" games, I was glad to see the Apple-II computer become available. With its high-resolution graphics capability, it is now possible to play more interesting games. Rocket Pilot is my first offering; in the months to come I plan to produce many more such action games — and already have several in the works.

The Game

Rocket Pilot is written for an Apple-II computer with at least 16K bytes of memory. The object of the game is very simple: Fly a rocket ship over a mountain and land on the other side. The flight is controlled by the game paddles (or a joystick, if you have one). As in other Lunar Lander games, you must descend gently in order for the landing to be success-








```
 190 IF DY>0 THEN DY=0
 200 IF FUEL THEN 100
 210 GOTO 905
 300 VTAB 24: TAB 15: PRINT "END OF SKY!";
 310 GOTO 905
 400 IF FUEL THEN 100
 410 DDX=0:DDY=N
 420 VTAB 24: TAB 15: PRINT "OUT OF FUEL";
 430 GOTO 110
 800 VTAB 24: TAB 15
 810 RESULT= ABS (2*DX)+ ABS (DY)
 820 IF RESULT<50 THEN PRINT "  PERFECT!  ";
 830 IF RESULT>=50 AND RESULT<100 THEN PRINT " VERY GOOD! ";
 840 IF RESULT>=100 AND RESULT<150 THEN PRINT "ABOUT AVERAGE";
 850 IF RESULT>=150 AND RESULT<200 THEN PRINT " VERY POOR  ";
 860 IF RESULT>=200 THEN 900
 870 FOR K=1 TO 1000: NEXT K
 890 GOTO 950
 900 POKE 50,127: VTAB 24: TAB 15: PRINT " C-R-A-S-H ";: POKE 50,2
 905 COL=1
 910 FOR K=1 TO 10
 920 POKE 18,COL: CALL 2048+256
 925 COL=1-COL
 930 PRINT "";: REM  "BELL"
 935 CALL 2048+260
 940 NEXT K
 950 FOR K=1 TO 500: NEXT K
 960 TEXT : CALL -936
 970 VTAB 10: TAB 15: PRINT "OK, LET'S"
 980 VTAB 12: TAB 13: PRINT "TRY IT AGAIN..."
 990 GOTO 20
1000 REM  DRAW LANDSCAPE
1050 SENSE=1
1100 YLOLD=60:YROLD=60
1210 FOR X=70 TO 99
1220 YR=YROLD+ RND (7)
1225 IF YR>159 THEN YR=159
1230 YL=YLOLD+ RND (7)
1235 IF YL>159 THEN YL=159
1240 POKE 16,X:Y=YR:YOLD=YROLD: GOSUB 1500
1250 POKE 16,139-X:Y=YL:YOLD=YLOLD: GOSUB 1500
1255 YROLD=YR:YLOLD=YL
1260 NEXT X
1310 POKE 18,1
1320 FOR X=39 TO 15 STEP -1
1330 POKE 16,X
1335 POKE 17,YL
1340 CALL 2048
1350 POKE 16,139-X
1355 POKE 17,YR
1360 CALL 2048
1370 NEXT X
1400 SENSE=-1
1405 HLOLD=YL:HROLD=YR
1410 FOR X=14 TO 0 STEP -1
1420 POKE 16,X
1425 HL=HLOLD- RND (7)
1430 Y=HL:YOLD=HLOLD: GOSUB 1500
1440 POKE 16,139-X
1445 HR=HROLD- RND (7)
1450 Y=HR:YOLD=HROLD: GOSUB 1500
1460 HLOLD=HL:HROLD=HR
1470 NEXT X
1480 RETURN
1500 REM
1530 FOR H=YOLD TO Y STEP SENSE
1540 POKE 17,H
1550 POKE 18,0: CALL 2048
1560 POKE 18,1: CALL 2048
1570 NEXT H
1590 RETURN
4000 VTAB 22: TAB 17: PRINT "    ";
4100 TAB 37: PRINT "    ";
4200 VTAB 22: TAB 17: PRINT DX/N;
4300 TAB 37: PRINT -DY/N;
4400 VTAB 23: TAB 10: PRINT "    ";
4500 TAB 37: PRINT "    ";
4600 VTAB 23: TAB 10: PRINT TIME/4;
4700 TAB 37: PRINT FUEL/N;
4800 RETURN
5000 CALL -936
5010 VTAB 10: TAB 10: PRINT "*** ROCKET PILOT ***"
5020 VTAB 13: TAB 19: PRINT "BY"
5030 VTAB 15: TAB 12: PRINT "ROBERT J. BISHOP"
5050 RETURN
```

ful. The difference is that your burn rate is determined by the position of the paddle controls, not by typing in numbers.

Each flight continues until you either make a successful landing, crash, run out of fuel or attempt to fly beyond the TV display area (in which case you run out of sky!). If you run out of fuel you lose control of the rocket, and it starts to fall under the influence of gravity. When you make a successful landing, the computer tells you how well you did.

Throughout the course of the flight, the bottom of the screen continually displays your horizontal and vertical velocities, elapsed time and remaining fuel.

The Program

Rocket Pilot is written partly in BASIC and partly in

machine language. Before entering the BASIC part, you should set LOMEM to 4096 (to leave room for the machine-language subroutines), and HIMEM to 8192 (to keep out of the high-resolution display buffer).

There are two sets of machine-language subroutines; one starts at 0800 (hex), and the other begins at 0900. These can be implemented by using either the Apple's mini-assembler or by typing in the hex codes via the monitor. (The latter is probably easier.)

Finally, the small table of hex numbers shown in Example 1 must be stored starting at hex location 0F00.

That's all there is to it. You are now ready to fly your own spaceship. If you don't want to type the whole program in by hand, cassette tapes of the game are available for $14.95 from: Computer Playground, 6789 Westminster Ave., Westminster CA 92683.

Many happy landings! ■

```
0F00:  07  14  10  10  10  10  10  10
0F08:  38  38  38  38  38  38  38  38
0F10:  38  38  7C  7C  EE  82
```

Example 1.

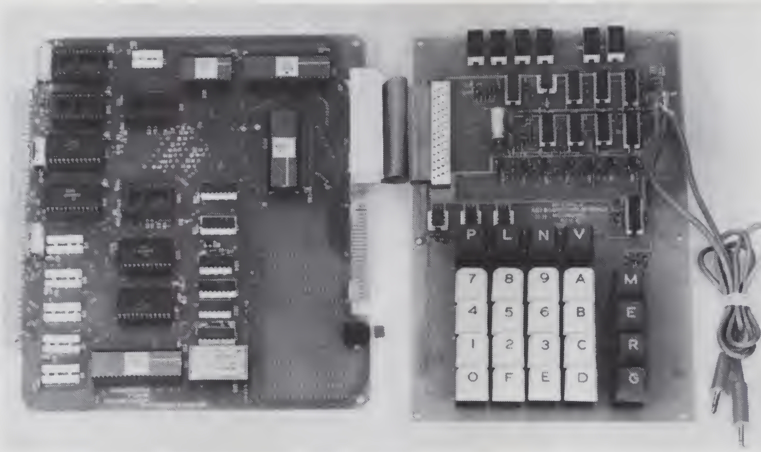
```
0800-   98          TYA
0801-   48          PHA
0802-   20 26 08    JSR   $0826
0805-   11 14       ORA   ($14),Y
0807-   91 14       STA   ($14),Y
0809-   68          PLA
080A-   A8          TAY
080B-   60          RTS
080C-   98          TYA
080D-   48          PHA
080E-   20 26 08    JSR   $0826
0811-   49 FF       EOR   #$FF
0813-   31 14       AND   ($14),Y
0815-   91 14       STA   ($14),Y
0817-   68          PLA
0818-   A8          TAY
0819-   60          RTS
081A-   98          TYA
081B-   48          PHA
081C-   20 26 08    JSR   $0826
081F-   31 14       AND   ($14),Y
0821-   85 13       STA   $13
0823-   68          PLA
0824-   A8          TAY
0825-   60          RTS
0826-   20 47 08    JSR   $0847
0829-   20 6D 08    JSR   $086D
082C-   A5 14       LDA   $14
082E-   18          CLC
082F-   65 16       ADC   $16
0831-   90 02       BCC   $0835
0833-   E6 15       INC   $15
0835-   85 14       STA   $14
0837-   A4 17       LDY   $17
0839-   B9 40 08    LDA   $0840,Y
083C-   A0 00       LDY   #$00
083E-   60          RTS
083F-   EA          NOP
0840-   01 02       ORA   ($02,X)
0842-   04          ???
0843-   08          PHP
0844-   10 20       BPL   $0866
0846-   40          RTI
0847-   A5 11       LDA   $11
0849-   0A          ASL
084A-   0A          ASL
084B-   29 1C       AND   #$1C
084D-   85 15       STA   $15
084F-   A5 11       LDA   $11
0851-   6A          ROR
0852-   6A          ROR
0853-   6A          ROR
0854-   6A          ROR
0855-   29 03       AND   #$03
0857-   05 15       ORA   $15
0859-   09 20       ORA   #$20
085B-   85 15       STA   $15
085D-   A5 11       LDA   $11
085F-   6A          ROR
0860-   29 E0       AND   #$E0
0862-   85 14       STA   $14
0864-   6A          ROR
0865-   6A          ROR
0866-   29 18       AND   #$18
0868-   05 14       ORA   $14
086A-   85 14       STA   $14
086C-   60          RTS
086D-   A9 00       LDA   #$00
086F-   85 16       STA   $16
0871-   A9 E0       LDA   #$E0
0873-   85 18       STA   $18
0875-   A9 20       LDA   #$20
0877-   85 19       STA   $19
0879-   A5 10       LDA   $10
087B-   85 17       STA   $17
087D-   A0 06       LDY   #$06
087F-   A5 17       LDA   $17
0881-   C5 18       CMP   $18
0883-   90 0B       BCC   $0890
0885-   E5 18       SBC   $18
0887-   85 17       STA   $17
0889-   A5 16       LDA   $16
088B-   05 19       ORA   $19
088D-   85 16       STA   $16
088F-   18          CLC
0890-   66 18       ROR   $18
0892-   66 19       ROR   $19
0894-   88          DEY
0895-   D0 E8       BNE   $087F
0897-   06 16       ASL   $16
0899-   A5 17       LDA   $17
089B-   0A          ASL
089C-   05 12       ORA   $12
089E-   C9 07       CMP   #$07
08A0-   90 04       BCC   $08A6
08A2-   E9 07       SBC   #$07
08A4-   E6 16       INC   $16
08A6-   85 17       STA   $17
08A8-   60          RTS
08A9-   8A          TXA
08AA-   48          PHA
08AB-   98          TYA
08AC-   48          PHA
08AD-   A9 00       LDA   #$00
08AF-   85 14       STA   $14
08B1-   A2 20       LDX   #$20
08B3-   86 15       STX   $15
08B5-   A8          TAY
08B6-   91 14       STA   ($14),Y
08B8-   C8          INY
08B9-   D0 FB       BNE   $08B6
08BB-   E6 15       INC   $15
08BD-   CA          DEX
08BE-   D0 F6       BNE   $08B6
08C0-   68          PLA
08C1-   A8          TAY
08C2-   68          PLA
08C3-   AA          TAX
08C4-   60          RTS
08C5-   EA          NOP
```

Subroutine 1.

```
0900-   A9 FF       LDA   #$FF
0902-   D0 02       BNE   $0906
0904-   A9 00       LDA   #$00
0906-   85 1F       STA   $1F
0908-   8A          TXA
0909-   48          PHA
090A-   98          TYA
090B-   48          PHA
090C-   A5 11       LDA   $11
090E-   48          PHA
090F-   A2 00       LDX   #$00
0911-   AD 01 0F    LDA   $0F01
0914-   85 1B       STA   $1B
0916-   AD 00 0F    LDA   $0F00
0919-   85 1A       STA   $1A
091B-   20 26 08    JSR   $0826
091E-   85 1C       STA   $1C
0920-   A9 80       LDA   #$80
0922-   85 1D       STA   $1D
0924-   BD 02 0F    LDA   $0F02,X
0927-   25 1D       AND   $1D
0929-   F0 02       BEQ   $092D
092B-   A9 FF       LDA   #$FF
092D-   85 1E       STA   $1E
092F-   A5 1C       LDA   $1C
0931-   24 1F       BIT   $1F
0933-   70 09       BVS   $093E
0935-   25 1E       AND   $1E
0937-   49 FF       EOR   #$FF
0939-   31 14       AND   ($14),Y
093B-   50 0D       BVC   $094A
093D-   EA          NOP
093E-   31 14       AND   ($14),Y
0940-   05 13       ORA   $13
0942-   85 13       STA   $13
0944-   A5 1C       LDA   $1C
0946-   25 1E       AND   $1E
0948-   11 14       ORA   ($14),Y
094A-   91 14       STA   ($14),Y
094C-   A5 1C       LDA   $1C
094E-   0A          ASL
094F-   10 04       BPL   $0955
0951-   A9 02       LDA   #$02
0953-   D0 05       BNE   $095A
0955-   0A          ASL
0956-   10 03       BPL   $095B
0958-   A9 01       LDA   #$01
095A-   C8          INY
095B-   85 1C       STA   $1C
095D-   66 1D       ROR   $1D
095F-   90 09       BCC   $096A
0961-   66 1D       ROR   $1D
0963-   E8          INX
0964-   C6 1A       DEC   $1A
0966-   D0 BC       BNE   $0924
0968-   F0 05       BEQ   $096F
096A-   C6 1A       DEC   $1A
096C-   D0 B6       BNE   $0924
096E-   E8          INX
096F-   E6 11       INC   $11
0971-   C6 1B       DEC   $1B
0973-   D0 A1       BNE   $0916
0975-   68          PLA
0976-   85 11       STA   $11
0978-   68          PLA
0979-   A8          TAY
097A-   68          PLA
097B-   AA          TAX
097C-   60          RTS
097D-   EA          NOP
097E-   EA          NOP
097F-   EA          NOP
0980-   EA          NOP
0981-   EA          NOP
0982-   EA          NOP
0983-   EA          NOP
0984-   EA          NOP
0985-   EA          NOP
0986-   EA          NOP
0987-   EA          NOP
```

Subroutine 2.

Terry Perdue
1470 Wilson Rd.
St. Joseph MI 49085



The Motorola MEK6800D2 Evaluation Kit.

Micro Maestro

a musical review of Motorola's MEK6800D2

For the last several years I have been noting with interest the ever-widening variety of applications for microprocessors. However, only recently did I decide that, as a design engineer, I had better stop procrastinating and start learning about this mysterious device, lest I be replaced by one.

I enrolled in a night class that began with number theory, proceeded through the workings of a hypothetical MPU, and finally covered the Motorola 6800 family. After learning the instruction set and addressing modes associated with the 6800, we began to write simple programs. As the programs became more complex, I began to wish that I had some means of running them to see if they would do what I intended. My employer agreed to purchase a trainer; the choice as to which one was left entirely up to me.

There are many trainer/evaluation systems available, and their features vary enough to make selection somewhat difficult. Having been introduced to the 6800 instructions, it seemed sensible to find a trainer using this chip. After a brief investigation of what was available, I came across a data sheet on the recently introduced Motorola MEK6800D2 Evaluation Kit. It seemed to have everything I needed to become more proficient in the use of the 6800 in particular, and programming in general, and the price of $235 was somewhat less than for others I had seen, so I ordered it.

Assembly

Upon opening the box, I found only a large three-ring binder full of documentation. Inside the binder, however, two of the "pages" contained circuit boards, with the associated components in plastic bubbles over each board. The assembly took about five hours and was aided by a manual containing pictures, parts placement drawings, and a page of construction hints.

The two boards, the Keyboard/Display Module and the Microcomputer Module, are joined by a 50-conductor ribbon cable. An edge connector on one end plugs onto the microcomputer circuit board, and the other end becomes permanently attached to the keyboard/display circuit board by means of another connector. Although the manual describes the installation of this connector, which is rather involved and requires a bench vise, the cable furnished had this connector already installed.

Sockets are provided for all of the 6800 family devices. These include the MC6800 MPU, an MCM6830 ROM containing the JBUG monitor, three MCM6810 RAMs (128 x 8), one of which is used as a scratchpad by JBUG, leaving 256 bytes for program storage, two MC6820 PIAs (Peripheral Interface Adapter), an MC6850 ACIA (Asynchronous Communications Interface Adapter), and the MC6871B crystal-controlled clock generator, operating at 614.4 kHz. The board has provisions for two additional RAMs, and two MCM68708 EPROMs (sockets provided), as well as for the buffers required to make the kit compatible with Motorola's EXORcisor. You may provide your own sockets for the remainder of the ICs, as I chose to do, but quality, low-profile types should be used.

The boards have mounting holes that allow spacers to be attached as feet, or the boards may be mounted into a chassis or enclosure. I found a large plastic box with carrying handle and integral hinge, and mounted the Microcomputer Module to the inside of the lid and the Keyboard/Display Module on long spacers to the bottom of the case. The long spacers provided room for a power supply beneath the board, making the unit self-contained and easily portable. The trainer requires a single 5 volt supply, and draws less than an Amp. I used a 6.3 volt filament transformer and bridge circuit, a 6000 uF electrolytic, and a 7805 regulator with heat sink. This should handle the two additional ROMs if added later, although if the use of the EPROMs is contemplated, plus and minus 12 volt supplies will also be required.

Features

The trainer provides a hexadecimal keyboard for program entry, and a six-digit

hexadecimal LED display. Eight function keys are also provided. They are labeled M, E, R, G, V, N, L and P. The M key allows examination and modification of any memory location. E stands for Escape, and allows an exit from a particular operation. R provides a look at the MPU's internal registers. G is used to go to any step in the user's program and begin execution, and is also used to single-step through memory when loading or reviewing a program. The V key is used to set breakpoints (up to five) for checking and debugging, while the N key traces through a program by executing one instruction at a time.

The L and P keys are used with one of the really nice features of the unit. The MEK6800D2 has built-in facilities for storing programs on audio cassettes. The trainer is simply attached to the microphone and external speaker jacks of the recorder. The first and last addresses of the program to be stored are entered, and the P (punch) key is pressed with the recorder in record mode.

To load a program from a cassette, the L button is pressed with the recorder in play mode. The user is notified of completion of either process by means of the prompt character (a dash in the left LED which means ready). After reloading a long program by hand, this feature is really appreciated.

Another thoughtful feature is a grid of plated holes on standard .1" x .3" centers which is provided on the microcomputer board to accommodate a large number of wire-wrap sockets for breadboarding additional circuitry.

Using the Trainer

While programming proficiency may take a long time, mastery of the trainer comes quickly. The manual that accompanies the kit explains its operation in detail, and leads the user through a sample program which illustrates the use of the various functions. In addition to the functions mentioned earlier, the JBUG monitor provides a useful routine for calculating the offset required for a branch instruction without manually counting addresses. This eliminates one of the most common programming errors.

While operation is quite straightforward, there are two things which I consider unhandy. When reviewing the MPU's registers using the R key, the value displayed as the location of the stack pointer is always seven less than the true value. When entering instructions or data, the program counter must be manually advanced with the G key. This adds many unnecessary steps to the loading of a program. These are only minor inconveniences, however.

The documentation is excellent. The assembly/operation manual contains a listing of the JBUG monitor with flowcharts for each of its functions, and schematic diagrams for each board. Also included are the System Design Manual, containing detailed data on the 6800 series devices, and the Programming Reference Manual, which fully describes the instruction set and addressing modes.

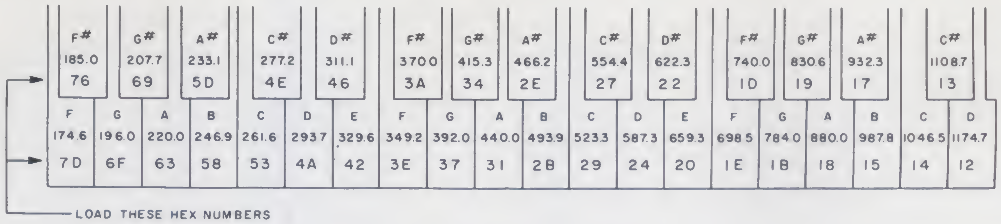


Fig. 1. Musical note-code chart.

Micro-maestro

Now that I've told you all about the trainer, let me give you an example of what fun it can be, aside from being instructive. After a few days of writing simple programs to gain confidence, I began to tire of watching the LED display for results, and decided to interface with the PIA. After learning how to address this versatile device, which can provide inputs and outputs in any desired software-controlled combination up to sixteen total, I wrote the following program. You can be sure that it didn't work the first time, or the second. But it does now, and as one of my first attempts at programming, I'm rather pleased with the result. It turns the trainer into a musical instrument which will play any tune you wish to load, using the note-codes given in the chart of Fig. 1 corresponding to the desired notes. The "instrument" is permanently tuned, due to

```
                                NAM       MUSIC
                                OPT       NOP,S
0000                            ORG       0
0000   7F    8005    REPLAY     CLR       $8005      CLEAR CRA-2 BIT
0003   7C    8004               INC       $8004      INCR DATA DIRECTION REG
0006   73    8005               COM       $8005      SET CR A-2 BIT
0009   8E    0025               LDS       *TEMP      POINT TO FIRST NOTE-1
000C   CE    08FF    RUNOTE     LDX       *$08FF     TIME PER NOTE
000F   33                       PUL   B              PULL NEXT NOTE FROM STACK
0010   5D                       TST   B              HAS LAST NOTE BEEN PLAYED?
0011   27    ED                 BEQ       REPLAY     IF NOT, CONTINUE
0013   F7    0025               STA   B   TEMP       STORE NOTE-CODE
0016   4C            TONLUP     INC   A              TOGGLES OUTPUT WHEN STORED
0017   F6    0025               LDA   B   TEMP       GET NOTE-CODE
001A   09            COUNT      DEX                  HOLD TONE FOR AWHILE
001B   27    EF                 BEQ       NUNOTE     LONG ENOUGH YET?
001D   5A                       DEC   B
001E   26    FA                 BNE       COUNT
0020   B7    8004               STA   A   $8004      TOGGLES OUTPUT
0023   20    F1                 BRA       TONLUP
0025   0001          TEMP       RMB       1
```

Fig. 2. Music program listing.

0026	53	0046	20	0066	42	0086	3E	00A6	2B	00C6	3E
0027	42	0047	29	0067	37	0087	58	00A7	37	00C7	4A
0028	53	0048	20	0068	42	0088	3E	00A8	2B	00C8	3E
0029	42	0049	29	0069	37	0089	58	00A9	37	00C9	4A
002A	53	004A	20	006A	3A	008A	3E	00AA	2B	00CA	24
002B	42	004B	29	006B	46	008B	58	00AB	37	00CB	3E
002C	53	004C	20	006C	3A	008C	3E	00AC	2B	00CC	24
002D	42	004D	29	006D	46	008D	58	00AD	37	00CD	3E
002E	53	004E	20	006E	3E	008E	3E	00AE	2B	00CE	29
002F	42	004F	29	006F	4A	008F	58	00AF	37	00CF	42
0030	53	0050	20	0070	3E	0090	3E	00B0	2B	00D0	29
0031	42	0051	29	0071	4A	0091	58	00B1	37	00D1	42
0032	37	0052	20	0072	3E	0092	37	00B2	2B	00D2	29
0033	42	0053	29	0073	4A	0093	4A	00B3	37	00D3	42
0034	37	0054	20	0074	3E	0094	37	00B4	2B	00D4	29
0035	42	0055	29	0075	4A	0095	4A	00B5	37	00D5	42
0036	37	0056	2B	0076	3E	0096	37	00B6	31	00D6	29
0037	42	0057	37	0077	4A	0097	4A	00B7	3E	00D7	42
0038	37	0058	2B	0078	3E	0098	37	00B8	31	00D8	29
0039	42	0059	37	0079	4A	0099	4A	00B9	3E	00D9	42
003A	22	005A	24	007A	3E	009A	31	00BA	37	00DA	29
003B	2B	005B	2B	007B	4A	009B	3E	00BB	42	00DB	42
003C	22	005C	24	007C	3E	009C	31	00BC	37	00DC	29
003D	2B	005D	2B	007D	4A	009D	3E	00BD	42	00DD	42
003E	20	005E	29	007E	3E	009E	2B	00BE	3A	00DE	2B
003F	29	005F	37	007F	4A	009F	37	00BF	46	00DF	31
0040	20	0060	29	0080	3E	00A0	2B	00C0	3A	00E0	37
0041	29	0061	37	0081	4A	00A1	37	00C1	46	00E1	3E
0042	20	0062	42	0082	3E	00A2	2B	00C2	3E	00E2	42
0043	29	0063	37	0083	4A	00A3	37	00C3	4A	00E3	4A
0044	20	0064	42	0084	3E	00A4	2B	00C4	3E	00E4	00
0045	29	0065	37	0085	4A	00A5	37	00C5	4A		

Fig. 3. The "Mystery Song."

the crystal-controlled clock, so you can play accompaniment on another instrument if you like. In addition, each time the song plays, the program turns the trainer into a different instrument, giving the song a new character each time it repeats. This is done by selecting different harmonic combinations from the PIA. The selected harmonics appear on edge connector J1, pins H, J and K (the three lowest order bits of the A port of the PIA). They are mixed through 10k resistors and coupled to an audio amplifier. If a mating edge connector is not available, note that there are plated-through holes into which the resistors may be soldered to avoid defacing the board connector. I found that the B plus bus gives less clock noise than the ground bus when used as a ground return for the amplifier.

The program listing is given in Fig. 2. Load the program through address 0025. Begin loading your chosen song at address 0026 and follow the last note-code with a 00 to identify the end of the song. Now press E, 0000, G, and out comes music! To stop the song, use the master reset on the Microcomputer Module; sometimes using the E key to stop it will alter a few addresses in this program. I don't know why. To alter the speed at which the song is played, change the 08 at address 000D to something else.

Fig. 3 provides a song sequence for you that gives the impression of two notes played together by rapidly alternating. I won't tell you the name of the song, but I'm sure you'll enjoy it. This program should be easily adaptable to other 6800 systems by following the flowchart in Fig. 4.

I realize that this may be an elementary approach to computer music for many readers, but one has to start somewhere. Playing with this trainer has sparked my interest to the point that I have my own 6800 system on order, and look forward to finding other uses for it.

My thanks to Doug Bonham and Bob Furtaw for bringing me this far. ■

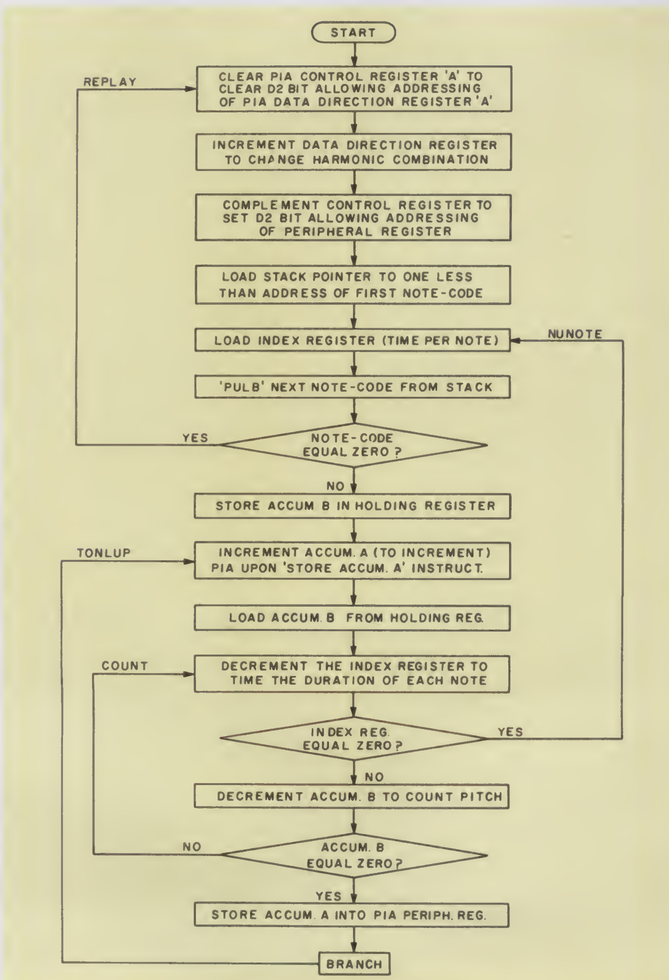


Fig. 4. Music program flowchart.

Volumes VI & VII are finally released!

Volume VI

Yes and it still contains what was previously advertised. A fully disk interactive business package with A/R, Inv., A/P, ledgers, tax totals, payroll records, more.

As a bonus it also contains the Users Manual for our Firmware Ledger package. These 100 extra pages contain report formats, file creation routines and our very powerful program ACBS1 used to create the powerful file structured data base.

$49.95

Volume VII

Here is that Chess program you have been waiting for as well as a disk interactive Medical Billing package with patient history file.

Also included is our disk interactive Word Processing package (revision Ø).

$39.95

Add $1.50/Vol. for U.P.S. and handling except to APO and PO addresses. Foreign orders add $8/Vol. for air shipment — US dollars only. No purchase orders over $50.

VOLUME I —	$24.95
VOLUME II —	24.95
VOLUME III —	39.95
VOLUME IV —	9.95
VOLUME V —	9.95

OUR SOFTWARE IS COPYRIGHTED AND MAY NOT BE REPRODUCED OR SOLD.

Due to the numerous copyright violations on our earlier volumes — until further notice we are offering a REWARD leading to the arrest and conviction of anyone reproducing our software in ANY way without our written permission. This includes diskettes, paper and magnetic tape, cassettes, records, paper copies, etc.

SRI SCIENTIFIC RESEARCH

220 Knollwood
Key Biscayne, Florida 33149

Phone orders (800) 638-9194

AVAILABLE AT MOST COMPUTER STORES

10% discount on purchases of entire set. Offer expires December 30, 1977

BANKAMERICARD master charge THE INTERBANK CARD

honored

Hal Chamberlin
29 Mead St.
Manchester NH 03104

Software Keyboard Interface

with a pittance of hardware!

I'll bet you're thinking, "Oh sure, another scheme using some obscure surplus keyboard that will be sold out by the time I get around to this project." Not so! This keyboard (manufactured by Datanetics Corp. of Fountain Valley CA) is offered by at least a half-dozen mail-order houses and is a current production item. But at $20 each (the most common price), these outfits are not doing us any favors; their cost is probably less than $10 each. An auxiliary keyboard the same style as the main unit is also available for less than $10 and can be used in this project for function keys, etc.

Why are these keyboards so cheap? The reason certainly is not lack of mechanical or electrical quality. They are unusually rigid one-piece construction of one-sixteenth-inch-thick Bakelite plastic, ribbed into a honeycomb form, with an overall depth of one-half inch. Each cell contains a contact arrangement with no fewer than four parallel contacts mounted inside a rugged plastic plunger. The contacts effectively reduce bounce and insure a long, error-free life. Finally, a keybutton is pressed onto the plunger, sealing the cell from dust and liquids.

One reason for the low cost is the one-piece base casting and cell structure. As I understand, the initial cost of the mold was borne by a huge quantity contract with Digital Equipment Corporation. However, the other reason is that the keyboard is devoid of any encoding electronics. This is the problem whose solution will be addressed in this article.

Besides the keyboard, the only other hardware this project requires is a single 74154 TTL integrated circuit (1-of-16 decoder), which costs less than two dollars, and some wire. Only four of the I/O port bits on the KIM's application connector are used, and even these may be used for other purposes when typing is not actually being done. Standard two-key rollover operation (which will be described later) is provided, and a full uppercase and lowercase ASCII character set is available. Even the repeat key works and has a programmable rate. The auxiliary keyboard is also supported with codes from its keys being identified by having the eighth bit set to a one. Even though some of the KIM's built-in keyboard circuitry is utilized, there is no conflict (with one small exception) between the built-in keypad and the new alphanumeric keyboard. A slight amount of additional circuitry using another IC may be added to have the break key function as an interrupt.

A software routine of approximately 350 bytes does all of the key scanning and code translation. This, in fact, is how the on-board KIM keypad is handled, with the difference being that the scanning software is in the KIM monitor ROM. If a code other than ASCII is desired, such as EBCDIC or Baudot, a translate table in the software may be easily altered. This table can be changed to suit different application programs, such as ASCII for running Tiny BASIC or Baudot for an automated RTTY application. The complete assembled and tested program is given at the end of this article.

Keyboard Scanning Theory

Nearly all keyboards in common use with more than a few keys use some kind of scanning logic to detect keyswitch closures, eliminate contact bounce, and generate unique key codes. In operation, scanning logic sequentially tests the state (up or down) of each individual key in the array. When a key is found in the down position, its code is determined and sent out. In order to avoid the code's being sent out more than once for each key depression, the scanning is stopped while the key is down and resumed when it is released. Typical scanning rates range from 20 to 500 complete scans per second of the approximately 60 keys in an average array.

Besides being a simple and inexpensive method of having a single logic circuit monitor the states of 60 individual keys, scanning also can cope with simultaneous key depressions. When someone is typing at substantial speed it is a common occurrence for more than one key to be down simultaneously. For example, consider rapid typing of the word THE. The T would first be pressed, followed shortly thereafter by a finger of the other hand pressing the H. Next the T would be released and the E would be quickly pressed with another finger of the same hand. Subsequently, the H would be released followed by the E, which completes the triad. A scanning keyboard would actually send the proper THE sequence to the computer, with no additional logic or buffer register required.

In order to understand how this works, let us examine the detailed sequence of events. Initially, no keys are pressed, and the scanning circuitry is running at full speed. When the T is pressed, the scanner eventually finds it, sends the T code and stops. As long as the T is held down, the scanner is stopped and testing the T key. While waiting for the T to be released, the typist presses the H, but the scanner is not aware of it. When the T is

finally released, the scanner takes off again but is immediately stopped when it sees that the H key is down. After sending the H code it waits for the H to be released, and so on.

If the typist is sloppy (or unusually fast) it is possible for even the E key to be pressed before the T is released, resulting in three keys being down simultaneously. In this situation, two keys are pressed while the scanner is waiting for the T key to be released. When scanning is resumed, two keys are down. The scanner will see the one that is closest to T in the scanning sequence and send that code next. The closest key might very well be the E, resulting in an error. This action on multiple key depressions is termed two-key rollover and is found on most computer terminals and other equipment used by casual typists. Some word-processing machines and other equipment used by professional typists have N-key rollover logic, which responds *only to the order* of key depression, regardless of how many keys are down simultaneously or the order in which they are released. Either special keyswitches or more complex scanning logic can be used to achieve N-key rollover. This keyboard interface is capable of N-key rollover with a more complex scanning program.

The scanning method can also easily take care of keyswitch contact bounce. When a closed contact is found, scanning is stopped, but sending of the code is delayed. If the contact should open during the delay, the closure is ignored and scanning is resumed without sending the code. If the momentary closure was really due to contact bounce, the key will be seen again on the next scan. If the closure is solid for the entire delay time, the code is sent. In addition, noise on contact opening may be rejected by requiring that the contact remain continuously open for a delay period before scanning is resumed. Typical values of debounce delay are one to five milliseconds.

Now, how is scanning circuitry typically implemented? One simple scheme for up to 64 keys would be to have an oscillator drive a 6-bit binary counter. The output of the counter would drive a decoder network having 64 separate outputs. All but one of the decoder outputs would be off, with the one on corresponding to the binary number in the counter. As the counter counts, each of the 64 decoder outputs would be turned on in sequence. For scanning a keyboard, each decoder output would be connected to one side of a keyswitch contact as shown in Fig. 1. The other sides of the contacts would all be connected together. This signal would be a zero except when a keyswitch was closed *and* that particular switch was addressed by the counter and decoder. With proper wiring between the decoder and the switch array, the 6-bit content of the binary counter while it is addressing a closed key can be the actual desired code of that key! Thus encoding is automatic with a scanning keyboard. Unfortunately, the shift and control keys of a typical keyboard complicate coding matters somewhat, but the basic concept is still valid.

Actually the scanning logic and switch wiring can be simplified greatly from the above conceptual model by arranging the keys in a matrix. Taking the same 64-key array, let us wire the keys in a matrix of eight rows and eight columns with a signal wire for each row and column. The contacts of a switch will be wired across each intersection, as shown in Fig. 2. Using the same 6-bit counter, let us connect three of the bits to a one-of-eight decoder and the other three bits to an 8-input multiplexer. A multiplexer is a logic circuit that has several signal inputs, some binary address inputs and one output. In operation, one of the signal inputs is logically connected to the output according to the binary code at the address inputs. The single output of the multiplexer is the addressed-key-closed signal as before. With matrix connection of the keys, the scanning logic grows in proportion to the square root of the number of keys, instead of directly.

As the scanning counter counts, the decoder activates one column of the matrix at a time and the multiplexer sequentially examines each row for a closed switch transferring the column signal over to a row. When a closed switch is found, the counter contains a unique code for the switch as before. Although it is still possible for this code to be the actual desired keycode, the scrambled key layout of a typical keyboard would make the matrix wiring quite messy. Typically a read only memory is used to translate the scramble code from the scanner into the end-use code the computer system needs. This same ROM also takes care of the shift and control keys, which are wired in directly.

Connection to the KIM

All of the previously described functions of scanning hardware can also be easily performed by software, along with an output and an input port. The most straightforward approach to simulate matrix scanning hardware would be to use an 8-bit

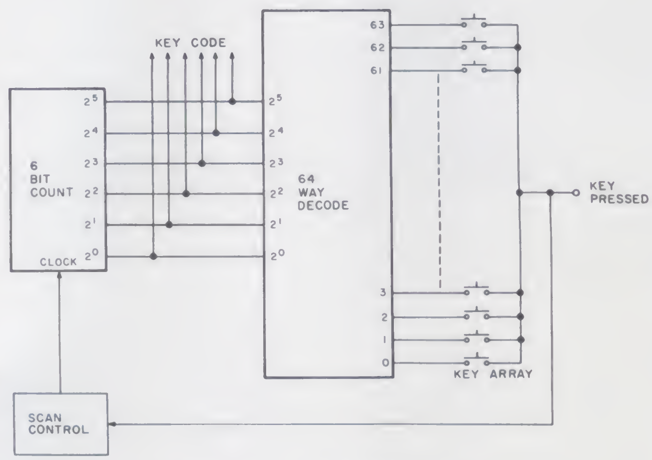


Fig. 1. Basic keyboard scanner.

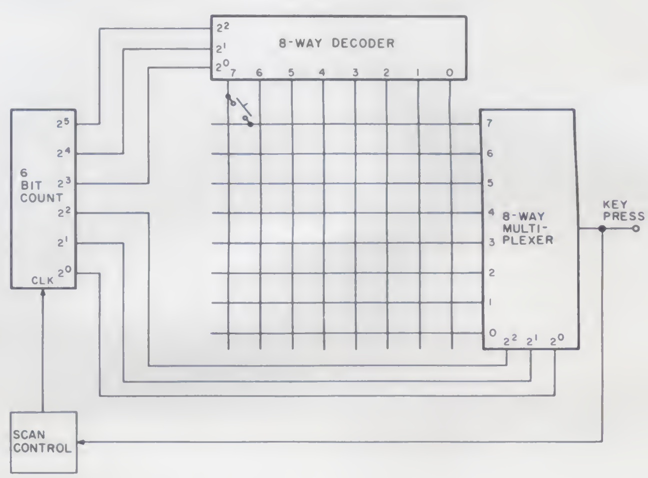
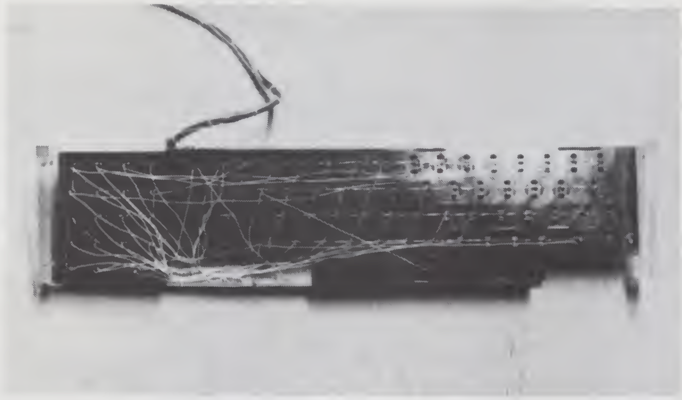


Fig. 2. Matrix keyboard scanner.



Keyboard point-to-point wiring.

output port with software to simulate the one-of-eight decoder and an 8-bit input port with software to simulate the 8-input multiplexer. The counter, of course, would be just a memory location that is incremented to perform the scanning. Unfortunately, in the case of the KIM this would utilize all of the built-in ports and then some.

A look at the KIM manual will reveal that much of the circuitry for the on-board keypay has signals brought out to the application edge connector. In particular, seven bits of an internal input port are available. These are connected internally to the on-board keypad and seven-segment displays, but when the KIM monitor is not running (user program running) they are completely free for use as an input port. Of course, when the monitor *is* in control, these inputs must not be driven by external circuitry, or interference with the keypad and display will result. If this port is connected to the rows of a key matrix and no keys are pressed, then nothing is driving the row wires; they are just hanging. Thus, when using the KIM monitor, one would not expect to be typing on the external keyboard so any interference is completely avoided.

At this point, one could use an 8-bit output port on the KIM to drive the key matrix and handle up to 56 keys without any interfacing circuitry. If you do not need the one full 8-bit port, and a limited character set (some missing symbols) is sufficient for your needs, then this can indeed be done. However, on my system the 8-bit port is connected to a digital-to-analog converter (for playing music) and two of the seven bits on the other port are motor controls for two cassette recorders. This leaves five bits for selecting the column to be scanned. The solution is to use four of these bits and an external 1-of-16 decoder to drive up to 16 columns. Combined with seven rows, up to 112 keys could be scanned.

Fig. 3 shows the connections to the KIM and the matrix hookup of the keys. Note that the optional 19-key keyboard is included. The arrangement of keys in the matrix was chosen mostly for simplicity of wiring, with proper coding taken care of with translation software. The one exception is the wiring of the 0-F keys on the auxiliary keyboard. They are in order with the 0 key in column 0, 1 key in column 1, etc. This would simplify a scanning routine that uses just those 16 keys. The 74154 decoder needs about 35 milliamps of +5 volt power. This should not strain any decent power supply for the KIM, but could be reduced to a mere 10 milliamps if a 74LS154 was substituted.

Note that the two shift keys are both wired into the matrix at row 3, column 15. The key labeled SHIFT on the auxiliary keyboard is intended to be relabeled and used for a less redundant function. The shift lock can be connected across the other two shift keys, but a problem arises in doing so. If it is left in the lock position when using the KIM monitor, there can be interference between the add-on keyboard and the KIM keyboard. If the shift-lock function is desired, and the requirement that it be unlocked before using the monitor is not judged to be bothersome, then the shift-lock key may be wired in.

Wiring the little tabs sticking out of the back of the keyboard should not be difficult. They are stiff enough and long enough to be wire-wrapped, too, if care is taken. Actually, this would be an ideal use of a Vector wiring pencil, which should get the job done in about 30 minutes. If hand wiring and soldering must be done, however, it is permissible to use bare bus wire for the row wiring and insulated wire for the columns. The purist can mount the 74154 IC in a socket on a piece of perfboard, but there is no reason that it cannot be glued to the bottom or side of the keyboard and wired directly.

The little circuit in Fig. 4 can be added to allow the Break key to be used as an interrupt. The KIM board would respond to this key in

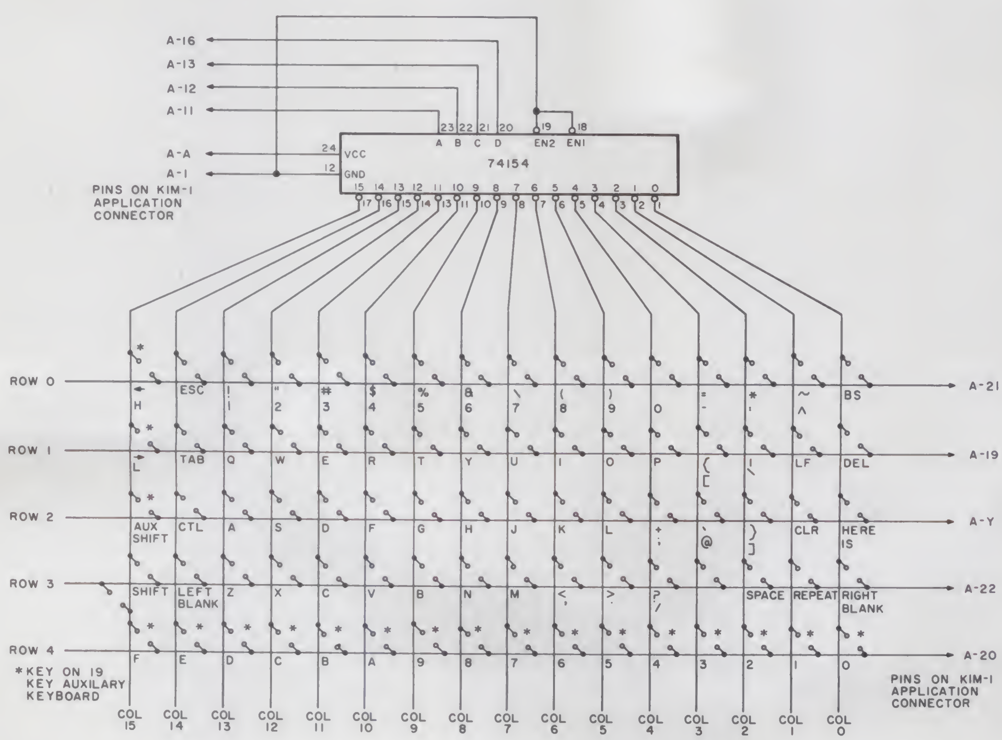
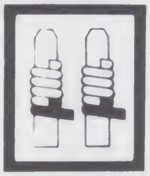


Fig. 3. Complete KIM-1 alphanumeric keyboard interface schematic.

"INDUSTRIAL"
WIRE WRAPPING TOOL

Model BW 520 is a battery operated wire-wrapping tool, also available in reversible models for wrapping and unwrapping operations. The rechargeable nickel cadmium battery is self contained in the handle of the tool. Also available with "Backforce" device to prevent overwrapping.

Uses any 24AWG thru 32AWG wrapping bits and sleeves.



OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 / (212) 994-6600 / Telex 125091

Model **BW520** **NEW**

BATTERY WIRE-WRAPPING TOOL

ONLY **$108.70**

BATTERIES INCLUDED

BIT & SLEEVE NOT INCLUDED

Model **BW928** **NEW**

BATTERY WIRE-WRAPPING TOOL

ONLY **$49.95**

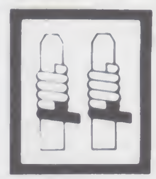
BATTERIES NOT INCLUDED

BIT & SLEEVE NOT INCLUDED

"INSTALLATION AND FIELD SERVICE"
WIRE WRAPPING TOOL

Model BW 928 is a battery operated wire wrapping tool, also available in reversible models for wrapping and unwrapping operations. Exceptionally useful for installation and field service work, or small production jobs. Available with "Backforce" device to prevent overwrapping.

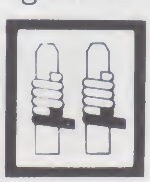
Uses any 24AWG thru 32AWG wrapping bits and sleeves.



OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 / (212) 994-6600 / Telex 125091

"HOBBY"
WIRE WRAPPING TOOL

For AWG 30, .025" (0.63mm) sq. post, "MODIFIED" wrap, positive indexing, anti-overwrapping device.



05

OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 / (212) 994-6600 / Telex 125091

Model **BW630** **NEW**

BATTERY WIRE-WRAPPING TOOL

ONLY **$34.95**

BATTERIES NOT INCLUDED

COMPLETE WITH BIT & SLEEVE

the same manner as the ST key on the built-in keypad and return to the monitor. However, if the nonmaskable interrupt (NMI) vector is changed at 17FA and 17FB, the interrupt could jump to a specific point in the user's program instead. The resistors, capacitor and 7413 Schmitt trigger IC debounce the break key to prevent multiple interrupts. The diode in series with the output simulates an open-collector output so that normal ST key operation is not affected. Preferably, the diode is a germanium type such as a 1N34 or 1N270, but a silicon unit will generally work OK.

Scanning Program

The program in Fig. 5 is the heart of the add-on keyboard system and is responsible for most of its features. Although shown assembled for locations 0200-035C (hexadecimal), it may be modified for execution anywhere by changing those locations marked with an underline in the object-code column. One temporary storage location is required on page 0. Its initial value when the keyboard is first used in a user program is not important, but thereafter it should not be bothered. The routine may be interrupted with no ill effects, but it is not reentrant (that is, it may not be called by an interrupt-service routine if it was itself interrupted) due to the temporary storage location just mentioned. This temporary location is at 00EE (just below the KIM reserved area) in the listing shown but may be easily moved elsewhere.

Using the program is quite simple. It is called as a subroutine whenever a character from the keyboard is needed. The contents of the registers when called are not important. When called, the routine waits until a key is pressed (except for code, shift or repeat). When a key is pressed, its code is loaded into the accumulator and a return taken. For maximum flexibility, the contents of the index registers are not disturbed by the routine.

Before you get into the program logic, perhaps a word should be said about the assembly language. The assembler used to prepare the listing is a modified version of the National Semiconductor IMP-16, which, in turn, is similar to the PACE assembler. In most respects, the syntax conforms to that recommended by MOS Technology. The major difference is that hexadecimal constants are denoted by X' instead of $. The use of a # before a constant or symbol specifies the immediate addressing mode. The assembler automatically distinguishes between zero page and absolute mode addressing according to the numerical magnitude of the address — zero page if between 0000 and 00FF and absolute otherwise. The various indexed and indirect addressing modes are represented in the same way as with the MOS Technology assembler.

The overall logic of the keyboard subroutine closely parallels that described for a hardware keyboard scanner. The first step when it is entered is to save the index registers on the stack. Next, the direction registers for the input and output port bits are set up. Note that only the direction bits for the port bits actually used are changed; the others are left unchanged.

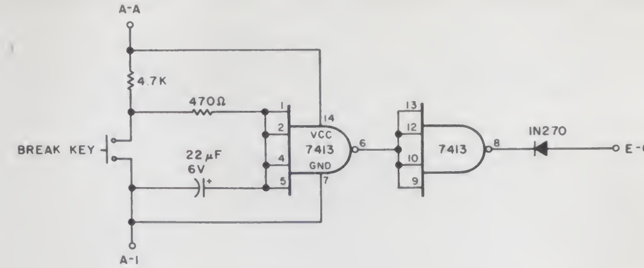


Fig. 4. Optional break-key interface.

When the subroutine is entered, an assumption is made that the last key pressed is still down. This is certainly a valid assumption since a return from the previous invocation of this subroutine occurred immediately when a key was pressed, and it is unlikely that processing of that character by the calling program took very long. ANKBT1 is the temporary storage location mentioned earlier. Functionally, it is equivalent to the counter in a hardware keyboard scanner. It always addresses a key in the matrix, and in this case it points to the key that was last pressed and had its code sent.

Thus, after saving the registers and setting up the ports, a loop is entered in which the keyboard routine is waiting for this last-pressed key to be released. While in this waiting loop, the status of the repeat key is continually interrogated. If the repeat key is continuously down while the last-pressed key is also continuously down for the repeat period, an exit is taken from the loop and the key code is sent again. Note that the repeat period, RPTRAT, is a parameter that may be changed; in this case it is set to 50 milliseconds, giving a moderately fast repeat rate of approximately 20 characters per second.

An internal subroutine, KYTST, is used to actually test the state of a key. It is used by loading the address of the key to be tested into the accumulator, and then calling it. When it returns, the carry flag will be on if the key is up, and off if it is down.

The other exit from this waiting loop, of course, is sensing that the last addressed key has been released. A debounce delay (DBCDLA) is included to insure that the key is interpreted to be up only when it has been continuously up for the debounce delay period. This will prevent noisy contacts from generating multiple characters.

At this point, scanning of the keyboard resumes.

Fig. 5. KIM-1 alphanumeric keyboard scan and encode routine.

```
                    .PAGE  'KIM-1 ALPHANUMERIC KEYBOARD SCAN AND ENCODE ROUTINE'
              ;     THIS SUBROUTINE SCANS AN UNENCODED KEYBOARD MATRIX CONNECTED
              ;     TO THE KIM-1 APPLICATION CONNECTOR.  USER PERIPHERAL PORT B
              ;     BITS 5 (MSB) THROUGH 2 (LSB) ARE CONNECTED TO A ONE-OF-16
              ;     DECODER (74154) WHICH DRIVES THE KEYSWITCH COLUMNS.
              ;     SENSING OF THE ROWS IS BY A PORTION OF THE KIM ON-BOARD
              ;     KEYBOARD CIRCUITRY WHICH USES SYSTEM PERIPHERAL PORT B BITS
              ;     0 - 4.
              ;     WHEN CALLED, THE ROUTINE SITS IN A LOOP WAITING FOR A KEY TO
              ;     BE PRESSED.  WHEN A KEY IS PRESSED (EXCEPTING SHIFT, CONTROL,
              ;     REPEAT), THE ROUTINE RETURNS WITH KEY CODE IN ACCUMULATOR.
              ;     BOTH INDEX REGISTERS ARE RETAINED.
              ;     THE ROUTINE IMPLEMENTS TRUE 2-KEY ROLLOVER, KEY DEBOUNCING,
              ;     AND REPEAT TIMING.  ONE RAM LOCATION IS REQUIRED, ITS INITIAL
              ;     CONTENT IS INSIGNIFICANT.

0000                .=      X'200        ; START PROGRAM AT LOCATION 0200 (HEX)

1740          SYSPA =       X'1740       ; SYSTEM PORT A DATA REGISTER
```

ok wire wrapping center ok

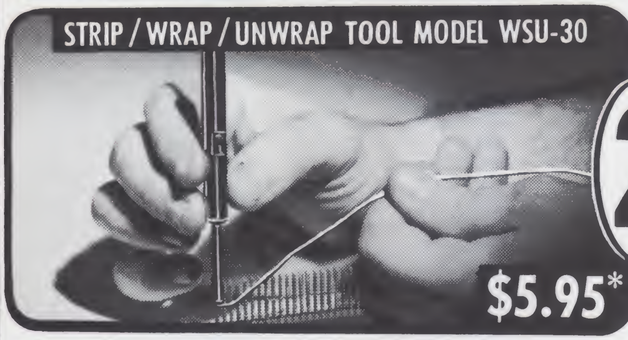
ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

1

BATTERY WIRE-WRAPPING TOOL MODEL BW-630

$34.95*

STRIP / WRAP / UNWRAP TOOL MODEL WSU-30



$5.95*

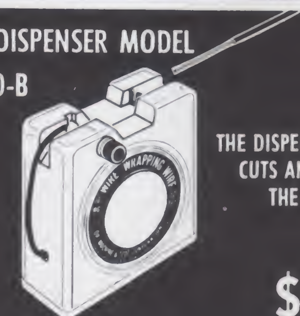
2

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

3

WIRE DISPENSER MODEL
WD-30-B



THE DISPENSER WHICH CUTS AND STRIPS THE WIRE

$3.45*

DIP IC INSERTION TOOL WITH PIN STRAIGHTENER
MODEL INS-1416

$3.49*

4

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

5

WHAT'S ? NEXT

* MINIMUM ORDER $25.00, SHIPPING CHARGE $1.00, N.Y. CITY AND STATE RESIDENTS ADD TAX

OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y 10475 ■ (212) 994-6600 ■ Telex 125091

```
     1741            SYSPAD   =      X'1741        ; SYSTEM PORT A DIRECTION REGISTER
     1702            USRPB    =      X'1702        ; USER PORT B DATA REGISTER
     1703            USRPBD   =      X'1703        ; USER PORT B DIRECTION REGISTER
     0032            RPTRAT   =      50            ; REPEAT PERIOD, MILLISECONDS
     0005            DBCDLA   =      5             ; DEBOUNCE DELAY, MILLISECONDS

     00EE            ANKBT1   =      X'EE          ; TEMPORARY STORAGE LOCATION ADDRESS


     0200 98         ANKB:    TYA                  ; SAVE THE INDEX REGISTERS
     0201 48                  PHA
     0202 8A                  TXA
     0203 48                  PHA
     0204 AD4117              LDA    SYSPAD        ; SET UP DATA DIRECTION REGISTERS
     0207 29E0                AND    #X'E0         ; SET SYSTEM PORT A BITS 4-0 TO INPUT
     0209 8D4117              STA    SYSPAD
     020C AD0317              LDA    USRPBD
     020F 093C                ORA    #X'3C         ; SET USER PORT B BITS 5-2 TO OUTPUT
     0211 8D0317              STA    USRPBD
     0214 A032                LDY    #RPTRAT       ; INITIALIZE REPEAT DELAY
     0216 A205       ANKB1:   LDX    #DBCDLA       ; INITIALIZE DEBOUNCE DELAY
     0218 207B02     ANKB2:   JSR    WA1MS         ; WAIT 1 MILLISECOND
     021B A5EE                LDA    ANKBT1        ; GET KEY ADDRESS LAST DOWN
     021D 208202              JSR    KEYTST        ; TEST IF ADDRESSED KEY STILL DOWN
     0220 B00C                BCS    ANKB4         ; JUMP IF UP
     0222 A931                LDA    #X'31         ; TEST STATE OF REPEAT KEY
     0224 208202              JSR    KEYTST
     0227 B0ED                BCS    ANKB1         ; LOOP BACK IF REPEAT KEY IS UP
     0229 88                  DEY                  ; DECREMENT REPEAT DELAY
     022A D0EA                BNE    ANKB1         ; LOOP BACK IF REPEAT DELAY UNEXPIRED
     022C F022                BEQ    ANKB7         ; GO OUTPUT REPEATED CODE
     022E CA         ANKB4:   DEX                  ; DECREMENT DEBOUNCE DELAY
     022F D0E7                BNE    ANKB2         ; GO TEST KEY AGAIN IF NOT EXPIRED

                     ;        PREVIOUS KEY IS NOW RELEASED, RESUME SCAN OF KEYBOARD

     0231 E6EE       ANKB5:   INC    ANKBT1        ; INCREMENT KEY ADDRESS TO TEST
     0233 A5EE                LDA    ANKBT1
     0235 C93F                CMP    #X'3F         ; SKIP OVER SHIFT
     0237 F0F8                BEQ    ANKB5
     0239 C92E                CMP    #X'2E         ; SKIP OVER CONTROL
     023B F0F4                BEQ    ANKB5
     023D C931                CMP    #X'31         ; SKIP OVER REPEAT
     023F F0F0                BEQ    ANKB5
     0241 A205                LDX    #DBCDLA       ; INITIALIZE DEBOUNCE DELAY
     0243 A5EE       ANKB6:   LDA    ANKBT1        ; TEST STATE OF CURRENTLY ADDRESSED KEY
     0245 208202              JSR    KEYTST
     0248 B0E7                BCS    ANKB5         ; GO TRY NEXT KEY IF THIS ONE IS UP
     024A 207B02              JSR    WA1MS         ; WAIT 1 MILLISECOND IF DOWN
     024D CA                  DEX                  ; DECREMENT DEBOUNCE DELAY
     024E D0F3                BNE    ANKB6         ; GO CHECK KEY AGAIN IF NOT EXPIRED

                     ;        TRANSLATE AND OUTPUT A KEY CODE

     0250 A6EE       ANKB7:   LDX    ANKBT1        ; GET BASIC ASCII CODE FROM TABLE
     0252 BCBD02              LDY    ANKBTB,X      ; INTO INDEX Y
     0255 A92E                LDA    #X'2E         ; TEST STATE OF CONTROL KEY
     0257 208202              JSR    KEYTST
     025A B006                BCS    ANKB8         ; SKIP AHEAD IF NOT PRESSED
     025C 98                  TYA                  ; CLEAR UPPER THREE BITS OF CODE IF
     025D 291F                AND    #X'1F         ; CONTROL PRESSED
     025F 4C7002              JMP    ANKB10        ; IGNORE SHIFT AND GO RETURN
     0262 A93F       ANKB8:   LDA    #X'3F         ; TEST STATE OF SHIFT KEY
     0264 208202              JSR    KEYTST
     0267 9004                BCC    ANKB9         ; SKIP AHEAD IF PRESSED
     0269 98                  TYA                  ; RETRIEVE PLAIN CODE FROM Y
     026A 4C7002              JMP    ANKB10        ; GO RESTORE REGISTERS AND RETURN
     026D BD0D03     ANKB9:   LDA    ANKBTB+80,X   ; FETCH SHIFTED CODE FROM TABLE
     0270 BA         ANKB10:  TSX
     0271 BC0201              LDY    X'102,X       ; RESTORE Y FROM STACK
     0274 9D0201              STA    X'102,X       ; SAVE CHARACTER CODE IN STACK WHERE Y WAS
     0277 68                  PLA                  ; RESTORE X
     0278 AA                  TAX
     0279 68                  PLA                  ; RESTORE CHARACTER CODE IN A
     027A 60                  RTS                  ; RETURN

                     ;        WAIT FOR ONE MILLISECOND ROUTINE

     027B A9C8       WA1MS:   LDA    #200          ; WAIT FOR APPROXIMATELY 1 MILLISECOND
     027D E901       WA1MS1:  SBC    #1
     027F D0FC                BNE    WA1MS1
     0281 60                  RTS

                     ;        KEY STATE TEST ROUTINE
                     ;        ENTER WITH ADDRESS OF KEY TO TEST IN ACCUMULATOR
                     ;        LEAVES BOTH INDEX REGISTERS ALONE
                     ;        SETS ANKBT1 TO ZERO IF ILLEGAL KEY ADDRESS AND TESTS KEY ZERO
```

Scanning is accomplished by incrementing ANKBT1 and calling KEYTST to look at the state of the newly addressed key. Note that the shift, code and repeat keys are specifically skipped in the scan sequence. Also note that another function of KEYTST is to detect an illegal key address and set ANKBT1 to zero if an illegal address occurs. Such an illegal address would normally occur after testing the last key in sequence, so the forced reset to zero would start another scanning cycle. If a key is found depressed, another loop is entered that verifies that it is continuously depressed for the debounce delay interval before it is declared to be really pressed.

Once a newly pressed key has been found (or the conditions for a repeated character have been satisfied), the key code must be generated. First, the current key address in ANKBT1 is translated into a plain unshifted character code by using it as an index into the first part of the code table. Next, the state of the control key is tested. If it is down, only the lower five bits of the translated code are retained, and an exit is taken. If control is up, then the shift key is tested. If it, too, is up, an exit is taken. If the shift key is down, however, the code is retranslated using the second part of the code table. Note that with a code like ASCII, with logical bit pairing (unshifted and shifted codes differ by only one bit), the second half of the code table might be replaced with *a little* more programming to make the adjustments necessary on shifted characters.

Finally, the two index registers are restored and a return taken. Note that some playing around with the stack was necessary to preserve the character code in A while the other registers were restored.

The key state test routine, KEYTST, takes a key address in A and tests if the corresponding key is pressed. After checking for a valid key

address, and correcting it if not, the lower four bits of the address are sent to the port bits that have the 1-of-16 column decoder connected to them. These four port bits are updated without affecting any of the other bits on the same port. After the column address is sent out, the remaining three upper bits of the key address are used to access a "mask table," which selects one of the five significant row input bits to test. Then the input port that senses the five rows is read and tested against the mask. The zero or nonzero result is transferred to the carry flag, which won't be destroyed during the register restore sequence.

The code translate table is divided into two parts. The first is for unshifted codes; the second is for shifted codes. The characters are in matrix-wise order, starting with row 0, column 0, going through the columns on row 0, proceeding to row 1, and so forth, ending with row 4, column 15. The table given is for ASCII on the main keyboard. The blank or oddly marked keys are assigned to useful ASCII control codes such as CR for the key marked CLR. The 0-F keys of the auxiliary keyboard become 80-8F for lowercase and 90-9F for uppercase. The remaining three auxiliary keys are assigned codes A0-A5. The table may be changed freely to reflect the user's choice of convenient control codes or to accommodate a completely different character code.

Building this keyboard interface for the KIM should prove to be a worthwhile one-evening project. Besides saving a substantial amount of money, it serves as a good learning tool and an excellent example of how software can substitute for hardware, offer a lot of extra features and still be easy to use. The basic concepts can be easily applied to expanding other low-cost microcomputer trainer boards.■

```



0282 C950      KEYTST:  CMP   #80             ; TEST IF LEGAL KEY ADDRESS
0284 9004               BCC   KEYTS1          ; SKIP AHEAD IF SO
0286 A900               LDA   #0              ; SET TO ZERO OTHERWISE
0288 85EE               STA   ANKBT1          ; UPDATE ANKBT1
028A 48        KEYTS1:  PHA                   ; SAVE A ON STACK
028B 8A                 TXA                   ; SAVE X ON STACK
028C 48                 PHA
028D AD0217             LDA   USRPB           ; CLEAR USER PORT B BITS 2-5
0290 29C3               AND   #X'C3
0292 8D0217             STA   USRPB
0295 BA                 TSX                   ; RESTORE KEY ADDRESS FROM STACK
0296 BD0201             LDA   X'102,X
0299 290F               AND   #X'0F           ; ISOLATE LOW 4 BITS OF KEY ADDRESS
029B 0A                 ASLA                  ; POSITION TO LINE UP WITH BITS 2-5
029C 0A                 ASLA
029D 0D0217             ORA   USRPB           ; SEND TO USER PORT B WITHOUT DISTURBING
02A0 8D0217             STA   USRPB           ; OTHER BITS
02A3 BD0201             LDA   X'102,X         ; GET KEY ADDRESS BACK
02A6 4A                 LSRA                  ; RIGHT JUSTIFY HIGH 3 BITS
02A7 4A                 LSRA
02A8 4A                 LSRA
02A9 4A                 LSRA
02AA AA                 TAX                   ; USE AS AN INDEX INTO MASK TABLE
02AB AD4017             LDA   SYSPA           ; GET SYSTEM PORT A STATUS
02AE 3DB802             AND   MSKTAB,X        ; SELECT BIT TO TEST AND SET CARRY FLAG
02B1 18                 CLC                   ; ACCORDINGLY
02B2 E900               SBC   #0
02B4 68                 PLA                   ; RESTORE X FROM STACK
02B5 AA                 TAX
02B6 68                 PLA                   ; RESTORE A FROM STACK
02B7 60                 RTS                   ; RETURN

02B8 01020408  MSKTAB:  .BYTE X'01,X'02,X'04,X'08    ; MASK TABLE FOR KEYTST
02BC 10                 .BYTE X'10

               ;        ASCII CHARACTER CODE TRANSLATE TABLE



                                                     ; UNSHIFTED SECTION
02BD 085E3A2D  ANKBTB:  .BYTE X'08,X'5E,X'3A,X'2D    ; BS  CARRET  :  -
02C1 30393837           .BYTE X'30,X'39,X'38,X'37    ; 0  9  8  7
02C5 36353433           .BYTE X'36,X'35,X'34,X'33    ; 6  5  4  3
02C9 32311BA0           .BYTE X'32,X'31,X'1B,X'A0    ; 2  1  ESC  (AUX H)
02CD 7F0A5C5B           .BYTE X'7F,X'0A,X'5C,X'5B    ; DEL  LF  BACKSLASH  [
02D1 706F6975           .BYTE X'70,X'6F,X'69,X'75    ; P  O  I  U
02D5 79747265           .BYTE X'79,X'74,X'72,X'65    ; Y  T  R  E
02D9 777109A1           .BYTE X'77,X'71,X'09,X'A1    ; W  Q  HT  (AUX L)
02DD 060D5D40           .BYTE X'06,X'0D,X'5D,X'40    ; HEREIS  CR  ]  @
02E1 3B6C6B6A           .BYTE X'3B,X'6C,X'6B,X'6A    ; ;  L  K  J
02E5 68676664           .BYTE X'68,X'67,X'66,X'64    ; H  G  F  D
02E9 736100A2           .BYTE X'73,X'61,X'00,X'A2    ; S  A  CTL  (AUX SHIFT)
02ED 00002000           .BYTE X'00,X'00,X'20,X'00    ; (RIGHT BLANK)  REPAT  SP
02F1 2F2E2C6D           .BYTE X'2F,X'2E,X'2C,X'6D    ; /  .  ,  M
02F5 6E627663           .BYTE X'6E,X'62,X'76,X'63    ; N  B  V  C
02F9 787A0000           .BYTE X'78,X'7A,X'00,X'00    ; X  Z  (LEFT BLANK)  SHIFT
02FD 80818283           .BYTE X'80,X'81,X'82,X'83    ; (AUX 0  1  2  3)
0301 84858687           .BYTE X'84,X'85,X'86,X'87    ; (AUX 4  5  6  7)
0305 88898A8B           .BYTE X'88,X'89,X'8A,X'8B    ; (AUX 8  9  A  B)
0309 8C8D8E8F           .BYTE X'8C,X'8D,X'8E,X'8F    ; (AUX C  D  E  F)

                                                     ; SHIFTED SECTION

030D 085E2A3D           .BYTE X'08,X'5E,X'2A,X'3D    ; BS  CARRET  *  =
0311 30292827           .BYTE X'30,X'29,X'28,X'27    ; 0  )  (  '
0315 26252423           .BYTE X'26,X'25,X'24,X'23    ; &  %  $  #
0319 22211BA3           .BYTE X'22,X'21,X'1B,X'A3    ; "  !  ESC  (AUX H)
031D 7F0A7C7B           .BYTE X'7F,X'0A,X'7C,X'7B    ; DEL  LF  VERTBAR  {
0321 504F4955           .BYTE X'50,X'4F,X'49,X'55    ; P  O  I  U
0325 59545245           .BYTE X'59,X'54,X'52,X'45    ; Y  T  R  E
0329 575109A4           .BYTE X'57,X'51,X'09,X'A4    ; W  Q  HT  (AUX L)
032D 060D7D60           .BYTE X'06,X'0D,X'7D,X'60    ; HEREIS  CR  }  GRAVEACCENT
0331 2B4C4B4A           .BYTE X'2B,X'4C,X'4B,X'4A    ; +  L  K  J
0335 48474644           .BYTE X'48,X'47,X'46,X'44    ; H  G  F  D
0339 534100A5           .BYTE X'53,X'41,X'00,X'A5    ; S  A  CTL  (AUX SHIFT)
033D 00002000           .BYTE X'00,X'00,X'20,X'00    ; (RIGHT BLANK)  REPAT  SP
0341 3F3E3C4D           .BYTE X'3F,X'3E,X'3C,X'4D    ; ?  >  <  M
0345 4E425643           .BYTE X'4E,X'42,X'56,X'43    ; N  B  V  C
0349 585A0000           .BYTE X'58,X'5A,X'00,X'00    ; X  Z  (LEFT BLANK)  SHIFT
034D 90919293           .BYTE X'90,X'91,X'92,X'93    ; (AUX 0  1  2  3)
0351 94959697           .BYTE X'94,X'95,X'96,X'97    ; (AUX 4  5  6  7)
0355 98999A9B           .BYTE X'98,X'99,X'9A,X'9B    ; (AUX 8  9  A  B)
0359 9C9D9E9F           .BYTE X'9C,X'9D,X'9E,X'9F    ; (AUX C  D  E  F)

0000                    .END
NO ERROR LINES
```

One Byte at a Time

with Morrow's I/O cassette board

Gary Alevy
1056 Fifth Avenue
New York NY 10028

The Morrow Cassette Interface I/O board is Altair-compatible and implements the Kansas City tape cassette standard for three cassette players with start/stop control. On the board there is also a serial and a parallel port. The board incorporates 1/2K bytes of RAM and 1/2K bytes of PROM containing routines for block-oriented tape operations, UART simulation and checksum calculation, among others. The firmware provided does not provide character incremental read/write utilizing the cassette start/stop capabilities of the interface. The routines contained in this article allow you to store data on tape a character at a time and conserve tape. This capability is useful in applications like automatic data logging, where events to be recorded occur at infrequent intervals or when one is typing a lengthy program source listing.

One might suppose that character incremental read/write could be accomplished with the firmware provided by setting length of the data block to be output equal to one. This is not possible because allowance must be made for the tape to come up to speed before a character is read. Users of the interface are also aware that a five-second leader is also recorded before each data block; it would be senseless to precede each single character with the five-second leader dictated by the firmware.

Read Subroutine

The read subroutine returns a single character in register A. On entry to the subroutine, registers are saved and the tape unit is turned on. The firmware subroutine TREAD is used to search for the sync signal and read the data character. The character read from the tape is temporarily stored in memory at RDATA. Next the tape unit is turned off and registers are

Address	Machine Code			Label	Assembly Language Op Code	Operand	Comments
FAD3	F5			MRCHR	PUSH	PSW	
	C5				PUSH	B	SAVE REGISTERS
	D5				PUSH	D	
	E5				PUSH	H	
	3E	03			MVI	A, 03H	TURN ON TAPE
	D3	04			OUT	4	UNIT 1
	06	82			MVI	B, 82H	SET TO READ UNIT 1
	0E	00			MVI	C, 00H	SET READ OPTION
	21	F3	FA		LXI	H,RDATADDR	CHARACTER ADDRESS
	11	01	00		LXI	D,LENGTH	1 CHARACTER
	CE	4D	80		CALL	TREAD	CALL ROM SUBROUTINE
	AF				XRA	A	TURN OFF TAPE
	D3	04			OUT	4	UNIT 1
	E1				POP	H	
	D1				POP	D	RESTORE REGISTERS
	C1				POP	B	
	F1				POP	PSW	
	3A	F3	FA		LDA	RDATADDR	PUT CHARACTER IN REG A
	C9				RET		EXIT MRCHR SUBROUTINE
FAF3				RDATA	DB		MEMORY LOCATION IN WHICH CHARACTER READ FROM TAPE IS TEMPORARILY STORED

Program A. Morrow read character routine.

restored. Finally, the character is transferred from memory and made available to the calling routine in the A register.

Write Subroutine

The write subroutine takes a character from the C register, starts the tape player, writes a sync signal, writes the character, then turns off the tape player. The duration of the sync bit should be adjusted to be the minimum value that gives reliable data read back with a given cassette player. Note that the character to be written is stored at the memory location called WDATA. Calls are made to two routines in the Morrow firmware FLUX and WTAPE. (These are explained in the documentation supplied with the interface.) Only the B and D register pairs are saved at the entry point labeled WCHAR; it is the user's responsibility to save other registers that are affected. ■

```
                                            Assembly Language
Address    Machine Code         Label      Op Code  Operand        Comments
                                                                   CHARACTER TO BE
                                                                   WRITTEN ENTERS IN
                                                                   REGISTER C
FABO       C5                   MWCHR      PUSH     B                SAVE SOME REGISTERS
           D5                              PUSH     D
           79                              MOV      A, C             STORE CHARACTER TO
           32   D2   FA                    STA      WDATA            BE WRITTEN
           06   03                         MVI      B, 03H         TURN ON TAPE UNIT 1
           21   50   00                    LXI      H, 0050H       SYNC DURATION       WRITE
           0E   0B                         MVI      C,0BH            SET PARAMETERS SYNC
           79                              MOV      A,C              FOR FLUX
           CD   00   80                    CALL     FLUX           CALL ROM SUBROUTINE
           0E   06                         MVI      C,06H            TO WRITE SYNC
           11   01   00                    LXI      D,LENGTH         1 CHARACTER
           21   D2   FA                    LXI      H,WDATADDR       CHARACTER ADDRESS
           CD   1C   80                    CALL     WTAPE          CALL ROM SUBROUTINE
           AF                              XRA      A                TURN OFF TAPE
           D3   04                         OUT      4                UNIT 1
           D1                              POP      D                RESTORE REGISTERS
           C1                              POP      B
           C9                              RET                     EXIT MWCHR SUBROUTINE
FAD2                            WDATA      DB                      MEMORY LOCATION
                                                                   IN WHICH CHARACTER
                                                                   TO BE WRITTEN ON
                                                                   TAPE IS TEMPORARILY
                                                                   STORED
```

Program B. Morrow write character routine.

Dennis E. Hamilton
1952 Baird Road
Penfield NY 14526

It's Your Hobby!

the critics be hanged

Since when are hobbyists required to justify their pastimes? Will we soon be driven to set aside spacecraft models and ensconce train layouts behind secret panels? Has it come to pass that legitimate hobbies are only those that enhance property values, save money, serve the community or earn tax deductions? If not, why is the computer hobbyist distressed by that jealous zinger, "But what does it *do*?"

Maybe we have it coming to us. Sure, there *are* dividends used as justification to onlookers. I can always mumble about "investment values and inflation hedges" when I'm found out as a closet stamp collector. There's some truth to that, but it isn't *why* I collect stamps. Yes, ham operators are handy in public emergencies, but is that why your rig can reach Antarctica by reflection from the moon? Of course not.

Perhaps coin collecting is the one hobby in America that requires no explanation. There's money in it. No justification required. At least that's how it looks. It must also look that way in countries having different ideologies — the hobby is banned!

The real stoics have to be model railroaders. After all, what they do with their constructions strikes most adults as pretty ridiculous. There's just no convincing appeal to utilitarian value or the work ethic. Yet train enthusiasts stand tall, taking it right on the chin. They also take solace in believing that, given the chance, most people couldn't resist an opportunity to sit at the controls, unobserved. Maybe computer hobbyists have an opening there too.

Modelers have it easier. The result, like art, needn't do *anything*. Even when the model is operational, as with a fully articulated, HO-scale, steam-locomotive model, it doesn't have to be running to be appreciated. Operation also has little to do with the joys, trials and frustrations of construction. Pride in a job well done, that personal satisfaction, is rekindled each time the modeler examines the result. Undiminished by lapse of time, the model — or a collection — captures for us the joy of its undertaking, just as recall of other pleasures, and disappointments too, is inspired by leafing through family snapshots.

Personal photographs share another property with hobbyist creations: They don't have the same significance to "outsiders." No wonder. Snapshots seldom inspire appreciation of intrinsic features. Instead, family pictures arouse memory of the depicted event (baby's first tooth). Any wedding photographer can confirm that clients buy catalogs of memories, not "good" pictures.

The keyword is *personal*. The value of our hobbies is internal to and inseparable from ourselves. Our hobbies are part of us, expressing our being. The value of hobbies is measured not by society, employers or even our most intimate relations, but by our own satisfaction in pursuing them.

So, when asked what your recently assembled microprocessor system does, don't be so quick to apologize. Our culture cherishes human spirit, diversity and folly too. Let us never need to account to anyone for that.

If *you* feel unsatisfied with having a machine and nothing to do with it, by all means do something! Satisfying applications of personal computers — large, small and hand held — can be found everywhere. The only requisite is *your* interest, imagination and perseverance. Just don't be surprised when what you choose for the computer to do strikes others as frivolous. Remember, it's *your* hobby.

Now, what about being *good* at your hobby? For that ambition, if you're plagued with it, you'll require an audience, a community of kindred spirits that recognizes hard-won achievement and provides criticism you must respect. Well, at least those critics ask more interesting questions than, "But what does it do?" ■

We're the memory house

Associated Electronics Company
1885 W. Commonwealth, Unit G
Fullerton, CA 92633
(714) 879 - 7541 879 - 7707

ORDER FORM

	QTY. KIT	QTY. ASSEMBLED
AE 8KS	☐	☐
AE 16 KPS	☐	☐
AE 1702A PRGM	☐	☐
BROCHURE	☐	

Name: ______________________

Address: ______________________

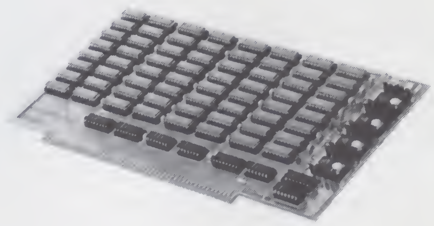
____

We accept: Checks • Money Orders • C.O.D. • Credit Cards
Master Charge #______________
BankAmericard #______________

A42

AE 8 KS (8K STATIC RAM)

- 8192 bytes of static memory.
- Access time: 500 nanoseconds.
- Memory chips are Signetics 21L02's.
- Memory allocation is in 8K increments to any location (switch selectable).
- Memory protection (full 8K) selectable with a switch.
- Fully socketed and buffered.
- Solder mask and character screen for ease of assembly.



Kit $189.00 Assembled $289.00

AE 16 KPS (16K PSEUDO-STATIC RAM)

- Runs with all popular processors including Polymorphics, Sol, Chromemco, ZPU, and others.
- Runs with Micropolis, North Star, Tarbell and other mass storage systems.
- Full speed memory chips (500 nanoseconds) with no wait states.
- All memory management functions on board. No software management required.
- Low power consumption.... 2.5 to 5 watts maximum!!!
- Headers allow allocating memory in 4K blocks to ANY LOCATION within the 65K 8080 or Z80 memory space. Allocation must be on 4K boundaries.
- Phantom, memory protect-unprotect, and other functions jumper selectable.
- This memory card is as DMA compatible as any other pseudo-static memory on the market. The DMA controller MUST HONOR ALL PROCESSOR SIGNALS!!! ASK DYNABYTE which DMA controllers they run with... then ask us!!!!
- Fully socketed and buffered.

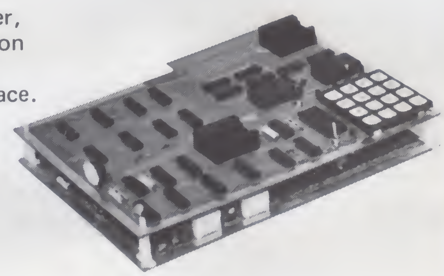
Kit $349.00 Assembled $449.00

AE 1702A PRGM (MANUAL 1702 PROGRAMMER)

Our new EPROM programmer has a vast array of improvements over the original design... with NO INCREASE IN PRICE!!!
THE ORIGINAL DESIGN has sold to John F. Kennedy Space Flight Center, Jet Propulsion Labs, Los Alamos Scientific Lab, Stanford Linear Acceleration Center, IBM, XEROX, 3M, and many others. The new design incorporates on-board 3 state buffers and two 16 pin headers for direct processor interface.

The new design also incorporates a new keyboard for increased reliability and improved function selection.

Write for additional details.



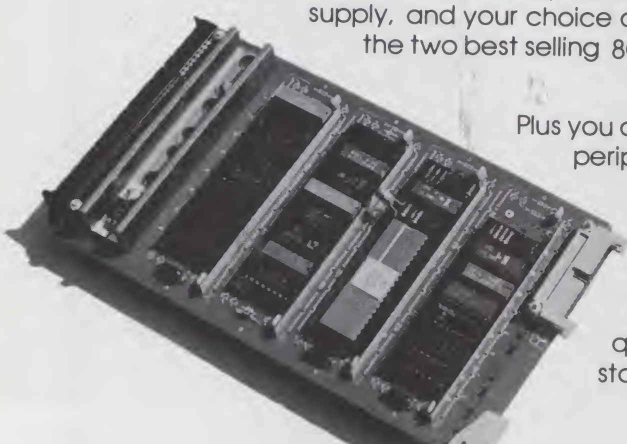
Kit $189.00 Assembled $299.00

WAVE MATE

(shown with Jupiter disk)

LOOKING BETTER ALL THE TIME

But we offer you more than just a beautiful new cabinet. The Wave Mate Jupiter IIA and Jupiter IIIA systems come to you fully assembled and tested, with backplane, plug-in terro-resonant power supply, and your choice of either 6800 or Z80 CPU modules. All for less than the price of the two best selling 8080 systems!



Plus you can choose from the fastest growing selection of memories and peripherals available from any manufacturer.

Our 2KB EPROM/4KB RAM/Serial interface module can start you on your way to high-quality, full capability, low cost personal computing. As your needs for computing power grow, add our video modules, our audio cassette interface, or even a high-speed matrix printer or floppy disk - all built with the same quality and dependability the Wave Mate name has come to stand for.

All Wave Mate products meet the highest quality industrial standards, with rugged construction unmatched by anyone. If you are serious about personal computing, call **Wave Mate.**

Send information on: ☐ Jupiter IIA
☐ Jupiter IIIA

NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ____________ ZIP ____________

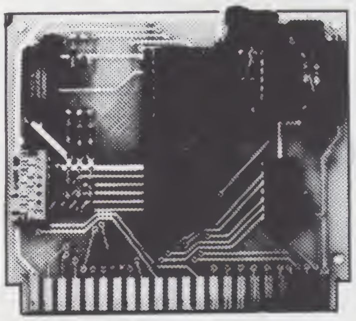
Wave Mate

WAVE MATE 1015 West 190th Street, Gardena, California 90248 Telephone (213) 329-8941
Dept. 25

W5

ELECTRONIC SYSTEMS
p.o. box 212 Burlingame CA
(408) 374-5984

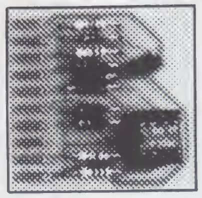
E21



UART & BAUD RATE GENERATOR

Part no. 101
- Converts serial to parallel and parallel to serial
- Low cost on board baud rate generator
- Baud rates: 110, 150, 300, 600, 1200, and 2400
- Low power drain +5 volts and -12 volts required
- TTL compatible
- All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity.
- All connections go to a 44 pin gold plated edge connector
- Board only $12.00; with parts $35.00

RS-232/TTL INTERFACE



Part no. 232
- Converts TTL to RS-232, and converts RS-232 to TTL
- Two separate circuits
- Requires -12 and +12 volts
- All connections go to a 10 pin gold plated edge connector
- Board only $4.50; with parts $7.00

DC POWER SUPPLY

Part no. 6085
- Board supplies a regulated +5 volts at 3 amps., +12,-12, and -5 volts at 1 amp.
- Board has filters, rectifiers, and regulators
- Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps.
- Board only $12.50

TAPE INTERFACE

Part no. 111
- Play and record Kansas City Standard tapes
- Converts a low cost tape recorder to a digital recorder
- Works up to 1200 baud
- Digital in and out are TTL
- Output of board connects to mic. in of recorder
- Earphone of recorder connects to input on board
- Requires +5 volts, low power drain
- Board $7.60; with parts $27.50
- No coils

Part no. 107

RF MODULATOR

- Converts video to AM modulated RF, Channels 2 or 3
- Power required is 12 volts AC C.T., or +5 volts DC
- Board $4.50; with parts $13.50

VISA master charge

To Order:

Mention part number and description. For parts kits add "A" to part number. Shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. Dealer inquiries invited. 24 Hour Order Line: (408) 374-5984.

LEGAL/BUSINESS FORUM

(from page 21)

brought any public actions against retailers for failing to comply with the provisions of the Presale Availability Rule. The reason stated is that it is difficult to determine if a product has been manufactured after January 1, 1977.

In my visits to computer stores, I have yet to see a single one complying with the rule. As a matter of fact, Lafayette and Radio Shack are the only stores I've seen that even make an attempt to comply. In those stores, if you look hard enough or ask for it, you will find a binder containing warranties.

SBCWA Repair Facility Requirements

If you think the Presale Availability Rule is a major revelation, this next one will blow your mind. The SBCWA mandates that if there is an express warranty accompanying goods sold in California, the manufacturers of such goods must maintain, in California, "sufficient service and repair facilities reasonably close to all areas where its consumer goods are sold" to carry out the terms of their warranties. Manufacturers may comply with this requirement by entering into warranty service contracts not exceeding one year with independent service and repair facilities. Providing facilities near Los Angeles and San Francisco will probably satisfy the "reasonably close" requirement.

If a manufacturer does not maintain repair facilities in California and has not authorized a representative to do so, you may return defective goods to either the retailer from whom you bought the goods or to any other retail seller of like goods of the same manufacturer. The retailer to whom the goods are returned may replace or repair at the retailer's option.

The SBCWA makes the manufacturer liable to the retailer who performs in-warranty service in an amount equal to what would be charged for nonwarranty work (which includes a reasonable profit). Retailers may recover treble damages if a manufacturer fails to reimburse the retailer. The implications here are astounding; I've yet to talk to a retailer who had received a single penny for performing in-warranty service.

The SBCWA also adds some teeth for you, the consumer. If you have gone to a retailer for warranty repairs and have not obtained satisfactory service, or if there is no retailer in California who sells like goods for the same manufacturer (i.e., if the manufacturer sells exclusively through mail order) you may secure the services of an independent service or repair facility. If you pursue this option, you are not responsible for the cost of repairs. The independent repair facility can only hold the manufacturer liable for such costs. Those costs encompass the actual costs of repair, including those for parts, transportation of the goods or parts, plus a reasonable profit. I have no doubt that many independent repair shops will emerge in the personal computing field.

A few comments about the repairment-repair provisions are in order: Those provisions are only applicable where the wholesale price to the retailer is $50. How do you know this? The SBCWA provides that the warranty must state that the independent repair option is available on all express warranties accompanying products with a wholesale price to the retailer of $50 or more. Again, I've yet to see a warranty that makes this required disclosure.

One more unique provision in the SBCWA is a tolling provision. Where any item costs the consumer $50 or more, the warranty period is tolled (extended) from the date the consumer delivers the goods for warranty repair until the goods are returned to the consumer.

The SBCWA contains additional disclosure requirements. An important addition to the MMWA requirements is that the consumer must be provided with the name and address of each repair facility in the state or with a toll-free number which will provide repair facility addresses for manufacturers who elect to maintain such facilities. The retailer must also provide a list of authorized repair facilities.

Can the MMWA and SBCWA Requirements be Avoided? The provisions discussed *cannot* be waived by any manufacturer or retailer. Any attempted waiver is void. The only out is simply to provide no warranty by making all sales on an "as is" basis. One well-known retailer is doing just that, providing a free "Service Contract" with the "as is" system. Coincidentally, the free service contract runs as long as the average warranty.

Remedies for Express Warranties

The UCC provides the same remedies for warranties whether implied or express. The SBCWA also provides identical remedies but allows treble damages in instances where an express warranty is breached. The MMWA does not broaden the scope of remedies available under the UCC and the SBCWA.

Perhaps the remedies available to the consumer seem slight compared to the magnitude of the requirements placed on manufacturers and retailers. However, the most important result of the warranty legislation is that the consumer is made aware of his warranty rights so he can enforce them. The SBCWA's repair-facility provisions are most important, as they place the onus on the manufacturer to establish repair facilities or to be liable to retailers or independents who provide repair facilities for warranty repairs. Presumably, this will be enforced by the repairer after the consumer has been satisifed. Nevertheless, in order to be effective at all, consumers must be aware of their rights and must take advantage of them.

NEW PRODUCTS

(from page 13)

new book, – *A Comprehensive Accounting System in BASIC*, by Dr. John Edwards, PhD.

Floppy diskettes will be available shortly, containing the Comprehensive Accounting System, for the following microcomputers: Alpha Microsystems, Altair, and Poly System 88.

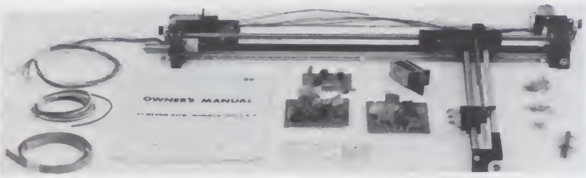
The book is priced at $149.95.

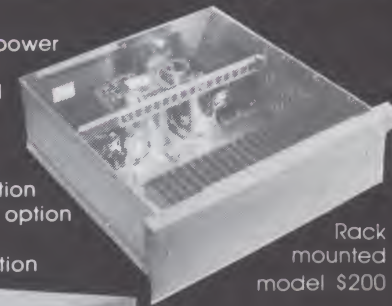
Microcomputer Business Systems, Inc., 1776 Plaza, 1776 East Jefferson Street, Rockville MD 20852.

Protect Your SOL Computer

An attractive and durable blue denim dust cover especially tailored for the Processor Technology Corporation SOL Terminal Computer is being offered by Curtis Electro Devices, Incorporated.

The cover comes in two standard versions: The first is embroidered with the legend "SOL COMPUTER," the second is lettered "MY COMPUTER." Custom legends are also available for a $5 additional cost. The standard cover is $14.95 postpaid from Curtis Electro Devices, Inc., Box 4090, Mountain View CA 94040.





The Prices? Low!
The Stock? On Hand!
The Deadline For Orders?

The Company? Computer Enterprises!

	Credit Card Price	Cash Price
Lear Siegler ADM-3A Kit	$764	**$735**
IMSAI 8080 with 22 Slots Kit	$614	**$590**
North Star Micro Disk System Kit	$623	**$599**
Polymorphic VTI/64 Kit	$182	**$175**
Comp. Enterp. Extender Board with connector	$ 13	**$ 12**
Comp. Enterp. Cable A for IMSAI	$ 14	**$ 13**
Seals 8K 250ns RAM Assmbld.	$238	**$229**
Vector Graphic 8K 450ns RAM Assmbld.	$234	**$225**
Vector Graphic PROM/RAM Kit	$ 83	**$ 80**
George Risk ASC II Keyboard Kit	$ 52	**$ 50**
Netronic's ELF II RCA COSMAC Microcomputer - wired & tested	$146	**$140**

ORDER NOW FOR THESE EXCITING **NEW PRODUCTS!**

North Star Horizon 1 Computer System	$1497	**$1439**
North Star Horizon 2 Computer System	$1871	**$1799**
IMSAI VIO-B Kit	$ 234	**$ 225**
IMSAI VIO-C Kit	$ 297	**$ 286**

Ask For Your FREE Computer Enterprises Catalog Today!

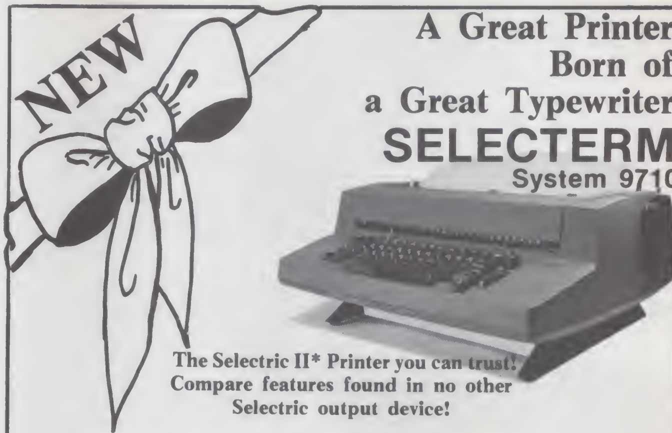
Shipping charges: $10 per CPU on larger units; $1.50 per kit. $2.00 min. per order.
Delivery is stock to 30 days on most items. Shipment is immediate for payment by cashier's check, money order or charge card. Allow 3 weeks for personal checks to clear. N.Y. State residents add approp. sales tax. Availability, prices and specs may change without notice.

Operating Hours:
M-W 10-5 E.S.T.
Th-F 10-9 E.S.T.
Closed Sat. & Sun.

Write or Call

computer enterprises™

P.O. Box 71
Fayetteville, N.Y. 13066 C33
Phone **(315) 637-6208** Today!



A Great Printer Born of a Great Typewriter
SELECTERM
System 9710

The Selectric II* Printer you can trust! Compare features found in no other Selectric output device!

- Brand new IBM Selectric II* Typewriter
- Fully converted for direct connection to any computer
- Special typing element for full ASCII and upper/lower case alphanumeric characters
- Direct connection to parallel or serial port at standard TTL level
- Responds to commands for backspace, vertical and horizontal tab, and bell
- Software in internal PROM handles all ASCII output conversions
- Can also be used as a standard typewriter
- Complete electronics package, cable sets and documentation
- Approved by IBM with warranty on typewriter and service contract
- Factory warranty on conversion package

M30

SELECTERM
micro computer devices
960 E. Orangethorpe, Bldg. F
Anaheim, California 92801
Telephone (714) 992-2270
"Innovators to the Microcomputer Industry"

Price $1650. Available only from dealers. Write for complete information.

*Registered trademark of IBM Corporation

"We may have the surplus electronics you need!"

Terminal/Keyboard
With Schematics
$250.00

- CRT Terminals
- Assemblies
- Peripherals
- Components

Tape Drives $800.00

Keyboards $40.00 to $60.00 (ASCII Encoded)

Equipment Cabinets $45.00 to $60.00 (19" With Fan)

Send for a free catalog or call
Bill Blaney, toll free 800 258-1036
in NH 603-885-3705
Come to our showroom

WORLDWIDE ELECTRONICS, INC.
10 Flagstone Drive, Hudson, New Hampshire 03051 W16

GLOSSARY

Rich Didday

Chemists have their secret codes ($Mg^{++} + 2e^{-} \rightarrow Mg^{\circ}$), English professors their puns (school daze), cowboys their boots, dirty old men their raincoats, Farrah Fawcett-Majors her hair, and . . . computer people their acronyms. Wherever computer people go, little clouds of *acronyms* surround them, some rubbing off on other people, some settling down in little piles, some continuing to swirl with a life of their own. Or so it seems.

Our acronyms make it difficult for the uninitiated to understand what's being said (maybe one of our purposes in using them), and yet make it easy and fast for insiders to communicate (our main excuse for using them).

What are they? My dictionary gives "radar" and "snafu" as examples of acronyms, or pseudo-words formed by selecting a letter (usually the first) or letters from the major words in a phrase. If I remember right, "radar" comes from RAdio Detection And Ranging. "Snafu" (which is the closest my dictionary has ever come to questionable taste) is an old (navy?) expression from WWII) oops, there's another one!). "Situation Normal, All Fouled Up" explains it well enough.

Why did computer people start using so many acronyms? Well, memory used to be very expensive, so early computer languages (and some not-so-early ones like BASIC) set severe limits on how many characters you could use in variable names, which encouraged an almost poetic terseness. And besides, names like EDSAC, ENIAC, UNIVAC and MANIAC must have seemed neat, tidy and futuristic to our founding fathers.

This glossary of acronyms includes (I hope) most of those computer hobbyists will normally encounter. It includes just acronyms, not abbreviations, so terms like "intercom," which is just short for "intercommunicator," are not included. Also, words (like PASCAL) that appear to be acronyms, but in fact are not originally based on some longer, meaningful phrase, are ignored. Finally, acronyms that are used only by specific manufacturers (like Motorola's DBE — Data Bus Enabled) have been weeded out.

If you know of an acronym in heavy use that's not listed here, please send it in. Then, write in demanding that we stop using so many of the blasted things.

A

A: Accumulator.
ac: Alternating Current.
ACIA: Asynchronous Communications Interface Adapter.
ACM: Association for Computing Machinery (a movement is afoot to quietly change the name to Association for CoMputing, thus making the name more reasonable without changing the all-important acronym).
ACR: Audio Cassette Recorder; Audio Cassette Recording system.
A/D: Analog to Digital.
ADC: Analog-to-Digital Converter.
ADP: Automatic Data Processing (businessmen who use computers also love acronyms).
AFIPS: American Federation of Information Processing Societies.
AI: Artificial Intelligence (the idea, not the existence of).
ALGOL: ALGOrithmic Language (a programming language).
ALU: Arithmetic and Logic Unit.
ANSI: American National Standards Institute (the name of this important group has gone through such stages as ASA, ASI and USASI — for the time being, it's ANSI).
AP: Artificial Perception.
APL: A Programming Language (a programming language originally defined by a modest man named Iverson).
ARPA: the Advanced Research Projects Agency (part of the "Department of Defense" that has funded major pieces of computer research and development).
ASAP: As Soon As Possible (when everybody wants everything).
ASCII: American Standard Code for Information Interchange.
ASR: Automatic Send/Receive (i.e., a TTY with paper-tape reader).

B

BASIC: Beginner's All-purpose Symbolic Instruction Code (who are they kidding here? — obviously they thought up the acronym first).
BCD: Binary Coded Decimal.
BIP: Binary Image Processor (one of the more successful truly parallel computers).
bit: BInary DigiT (this is really an acronym — "byte" and "nybble" aren't).
BPI: Bits Per Inch.
BPS: Bits Per Second.
BOT: Beginning Of Tape.

C

C: Capacitor; Controller; Clock.
CAD: Computer Aided Design.
CAI: Computer Assisted Instruction.
CAM: Content Addressable Memory.
CAR: Contents of the Address Register (see LISP).
CBEMA: Computer and Business Equipment Manufacturers Association (an important lobbying group for big computer manufacturers).
CCD: Charge Coupled Device.
CCR: Condition Code Register (what holds the status flags).
CDR: Contents of the Decrement Register (see LISP).
CFL: Context Free Language.
C^3L: Complementary Constant Current Logic.
CM: Central Memory; Community Memory (terminals on the street and in stores for public use).
CMOS: Complementary Metal-Oxide Semiconductor logic.
COBOL: COmmon Business Oriented Language (a programming language).
COM: Computer Output/Microfilm.
CP: Central Processor; Clock Pulse.
CPS: Characters Per Second.
CPU: Central Processing Unit.
CR: Carriage Return.
CRC: Cyclic Redundance Check.
CROM: Control Read-Only Memory.
CRT: Cathode Ray Tube (a fancy name for a TV picture tube).
CS: Computer Science; Check Sum.
CSL: Context Sensitive Language.
CYBORG: CYBernetic ORGanism (a six-million-dollar way to sell old Superman plots to TV viewers in the 1970s).

D

DA: Data Acquisition.
D/A: Digital to Analog.
DAC: Digital-to-Analog Converter.
DATACOM: DATA COMmunications.
dB: DeciBel (yes, named for A G Bell).
DBM: Data Base Management.
DBMIS: Data Base Management Information System.
dc: Direct Current (the kind Edison wanted).
DIP: Dual In-line Package (what most IC chips come in).
DIS: Distributed Information (processing) system.
DMA: Direct Memory Access (what you use when your CPU is too slow to handle the situation).
DOS: Disk Operating System.
DP: Data Processing.

dpdt: Double Pole, Double Throw (a kind of switch).
dpst: Double Pole, Single Throw (another kind of switch).
DRO: Digital ReadOut.
DSW: Device Status Word.
DTL: Diode-Transistor Logic.
DVM: Digital VoltMeter.

E

EBCDIC: Extended Binary-Coded-Decimal Interchange Code (IBM's alternative to ASCII, which uses all eight bits).
ECL: Emitter Coupled Logic.
EDC: External Device Code.
EDP: Electronic Data Processing.
EE: Electrical Engineering (pronounced "double E").
EEROM: Electrically Erasable Read-Only Memory.
EFL: Emitter Follower Logic.
EFT: Electronic Funds Transfer (basis of our monetary system in the future if the banks can figure out how to make a profit on it).
EIA: Electronics Industries Association.
EMI: ElectroMagnetic Interference.
EOF: End Of File.
EOM: End Of Message.
EOR: End Of Record.
EOT: End Of Tape.
EOTEWSWEO: End Of The Entries Which Start With "End Of."
EPROM: Erasable Programmable Read-Only Memory.
ET: Edge-Triggered.

F

FDM: Frequency Division Multiplexing.
FET: Field-Effect Transistor.
F-F: Flip-Flop.
FIFO: First In, First Out (a *queue* data structure).
FM: Frequency Modulation.
FORTRAN: FORmula TRANslator (a programming language).
FSA: Finite State Automation.
FSK:Frequency Shift Keying.

G

GIGO: Garbage In, Garbage Out (an old joke you set up by explaining FIFO and LIFO first).
GPSS: General Purpose Simulation System (a programming language/system).

H

HAL: Heuristically programmed ALgorithmic computer (2001).
HCD: Hard Copy Device.
HIPO: Hierarchy/Input-Process-Output (an IBM(?) acronym related to structured programming).
HLL: Higher-Level Language (i.e., not machine or assembly language).
HSM: High-Speed Memory.
HTL: High-Threshold Logic.

I

IAL: International Algebraic Language (a programming language now known as ALGOL 58).
IBM: The name of a computer company, formed by shifting each letter of HAL by one.
IC: Integrated Circuit.
ICE: In-Circuit Emulator.
ICS: Information and Computer Science.
IEEE: Institute of Electrical and Electronics Engineers.
IF: Intermediate Frequency.
IFIPS: International Federation of Information Processing Societies.
IL: Intermediate Language.
I^2L: Integrated Injection Logic.
I/O: Input/Output.
IR: Instruction Register.
IRED: InfraRed Emitting Diode.
IS: Information Science.
ISAM: Indexed Sequential Access Method.

J

JCL: Job Control Language (the thing you use on big computer systems to get an obscure error message telling you in incredibly abstruse terms why what you wanted to happen didn't).
JOVIAL: Joule's Own Version of IAL (a computer language).
JSR: Jump to SubRoutine.

K

K: 1024_{10}.
KB: KeyBoard; KiloByte; KiloBaud; *Kilobaud*.
KSR: Keyboard Send/Receive (e.g., a TTY with just keyboard and printer, no automatic input device).
KWIC: KeyWord In Context.

L

LASCR: Light-Activated Silicon-Controlled Rectifier.
LCD: Liquid Crystal Display (the kind most digital watches come with this season).
LDA: LoaD Accumulator (an instruction on most every computer ever made or thought of).
LED: Light Emitting Diode (the kind of display that most watches came with last season).
LF: Low Frequency.
LIFO: Last In/First Out (a push-down stack type of organization).
LISP: LISt Processing language (a computer language).
LP: Line Printer; Linear Programming.
LPM: Lines Per Minute.
LSB: Least Significant Bit; Least Significant Byte.
LSD: Least Significant Digit.
LSM: Low Speed Memory; Large Scale Memory (a big disk or maybe even high-speed mag tape).
LSTTL: Low power Schottky TTL.

M

MACLFAC&CO: Meaningless Acronym Composed Largely For Acronym Composers & Cerebral Orangutans.
MAD: Michigan Algorithm Decoder (a programming language).
MAL: Macro Assembly Language.
MAR: Memory Address Register.
MBM: Magnetic Bubble Memory.
MDS: Microprocessor Development System.
MIS: Millions of Instructions per Second (a coarse measure of a computer's power).
modem: MOdulator/DEModulator.
MOS: Metal Oxide Semiconductor technology.
MOSFET: Metal Oxide Semiconductor Field-Effect Transistor.
MPU: MicroProcessing Unit.
MPX: MultiPleX.
MSB: Most Significant Bit; Most Significant Byte.
MSD: Most Significant Digit.
MSI: Medium Scale Integration.
MTF: Mean Time to Failure ("mean" in the sense of "average").
MTBF: Mean Time Between Failures (doesn't quite sound as good in ad copy as MTF, but means the same thing).
MUX: MUltipleXer.

N

NAND: Not AND.
NC: Normally Closed; Numerically Controlled.
N/C: Numerical Control (used in the machine-tool industry to refer to the use of computers to control machines).
NIL: .
NMOS: n-channel MOS.
NO: Normally Open.
NOP: No OPeration.
NOR: Not OR.
NRZ: Non-Return to Zero.
NRZI: Non-Return to Zero Inverted.

O

OCR: Optical Character Recognition; Optical Character Reader.
OEM: Original Equipment Manufacturer (what to be if you want big discounts).
OR: Operations Research.
OS: Operating System.

P

PAM: Pulse Amplitude Modulation.
PC: Program Counter; Printed Circuit; Punched Card.
PCB: Printed Circuit Board.
PCM: Pulse Code Modulation.
PDM: Pulse Duration Modulation.
PDS: Program Development System.
PFM: Pulse Frequency Modulation.
PIA: Peripheral Interface Adapter.
PIO: Parallel Input/Output.
PIV: Peak Inverse Voltage.
PLA: Programmable Logic Array.
PLL: Phase-Locked Loop.
PL/I: Programming Language one (that's not an "I," it's a Roman numeral one — and that's how IBM says it should be written; not PL/1).
PL/M: Programming Language for Microprocessors.
PMOS: p-channel MOS.
POS: Point Of Sale (electronic/computer-controlled cash register).
PP: Peripheral Processor.
P-P: Peak-to-Peak.
PR: Pattern Recognition.
PSG: Phrase-Structure Grammar (e.g., the kind of grammar used to define ALGOL).
PSW: Program Status Word; Processor Status Word.
PSK: Phase-Shift Keying.
PTP: Paper-Tape Punch.
PTR: Paper-Tape Reader; PrinTeR.

Q

QA/4: Question Answerer version 4 (a programming language *cum* question-answering system — included mainly because I couldn't think of any other acronym that starts with Q).

R

RALU: Register, Arithmetic, and Logic Unit.
RAM: Random Access Memory (don't forget, the acronym RAM doesn't apply to ROM, even though ROM is random access

memory).
RC: Resistive, Capacitive circuit (i.e., an electrical circuit made up of interconnected resistors and capacitors only).
R/C: Remote Control.
R&D: Research and Development (a good thing to say you're doing if you want a government grant).
RF: Radio Frequency.
RFI: Radio Frequency Interference (hold a radio close to your CPU board and listen to some computer-generated RFI as you change the dial).
RJE: Remote Job Entry.
RMS: Root Mean Square (just what it says — take the square root of the average of the squares of the values under consideration).
RO: Read Only.
ROM: Read-Only Memory.
RPG: Report Program Generator (sort of a programming language).
RPN: Reverse Polish Notation (a parenthesis-free notation devised by the Polish mathematician Lukasiewicz).
RTE: Real-Time Executive (a kind of systems program).
RTL: Resistor-Transistor Logic.
RTOS: Real-Time Operating System.
R/W: Read/Write.
RWM: Read/Write Memory (a less pronounceable but more meaningful alternative to RAM).
RZ: Return to Zero.
RZI: Return to Zero Inverted.

S

SCR: Silicon Controlled Rectifier.
S/F: Store and Forward.
SIAM: Society for Industrial and Applied Mathematics.
SIMULA: SIMUlation LAnguage (a programming language).
SIO: Serial Input/Output.
SN: Semiconductor Network.
S/N: Signal-to-Noise Ratio.
SNOBOL: StriNg-Oriented symBOlic Language (a family of programming languages).
SOA: State Of the Art (where just about every manufacturer claims his products are).
SOM: Start Of Message.
SOP: Standard Operating Procedure.
SP: Structured Programming (a religion).
spdt: Single Pole Double Throw (a kind of switch).
spst: Single Pole Single Throw (another kind of switch).
SR: Shift Register.
SSI: Small-Scale Integration (the old days).
STL: Schottky Transistor Logic; Synchronous (MOS) Transistor Logic.
SWR: Standing Wave Ratio.

T

TDM: Time Division Multiplexing.
TELEX: TELEtypewriter eXchange service.
TIP: Terminal Interface Processor.
TRAC: Text Reckoning And Compiling (a programming language).
TS: Tough Situation.
TSS: Time-Sharing System.
TTL: Transistor-Transistor Logic.
TTY: TeleTYpewriter.
TWS: Translator Writing System.
TVI: TeleVision Interference (your neighbor seeing your game of Star Trek on top of Mike Wallace).
TVT: TeleVision Terminal.

U

UHF: Ultra High Frequency.
UJT: Uni-Junction Transistor.
UPC: Universal Product Code.
UPS: Uninterruptible Power Supply.
USART: Universal Synchronous/Asynchronous Receiver/Transmitter.
USPO: an almost-defunct government agency.

V

VCO: Voltage Controlled Oscillator (part of a PLL).
VM: Virtual Memory; Virtual Machine; VoltMeter.
VMA: Valid Memory Address; Virtual Memory Address.
VOM: Volt-Ohm Meter.
VTVM: Vacuum Tube VoltMeter.

W

WCS: Writable Control Store.
WE: Write Enable.
WOM: Write-Only Memory (don't everybody laugh at once).
WS: Working Store; Working Storage; Work Space.

X

XOR: eXclusive OR.
Xmitter: Transmitter.
Xtal: Crystal. (This and Xmitter aren't exactly acronyms, but at least they start with X.)

Z

Z/D: Zero Defects.

Sources

M. A. Cunningham and H. J. Gray, Design and test of a cognitive model, *Int. J. Man-Machine Studies*, 6, 1974, pp. 49-104.
Rich Didday, *Home Computers: 2^{10} Questions and Answers,* Volumes 1 and 2, dilithium Press, 1977.
Frank Kaiser, "Tired of decoding alphabet soup? Try this acronym dictionary," *EDN*, May 20, 1977, pp. 101-105.
The LSI-11 Microcomputer Glossary, Digital Equipment Corporation, 1976.
J. E. Sammet, *Programming Languages: History and Fundamentals*, Prentice-Hall, 1969.
C. J. Sippl and D. A. Kidd, *Microcomputer Dictionary and Guide,* Matrix Publishers, 1976.

KB CALENDAR

Atlanta GA

Papers are invited for presentation at the 16th Annual Convention of the Association for Educational Data Systems, Atlanta GA, May 15-19, 1978. Papers are solicited in all categories of educational use of computers. Papers submitted will be reviewed by a panel, and authors of those accepted will be invited to present their papers and to have them published in the proceedings.

Judges will select an outstanding paper from each category, and a panel will select a best paper on the basis of content, presentation and overall quality to receive a $500 cash award. For further information, contact: Dr. James E. Eisele, Office of Computing Activities, University of Georgia, Athens GA 30602.

Anaheim CA

The 1978 National Computer Conference will feature a Personal Computing Festival June 6-8 at the Disneyland Hotel complex in Anaheim CA. Hobbyists, consumers, students and computer professionals are invited to participate in the Festival and in the National Computer Conference June 5-8 in the Anaheim Convention Center. Both one-day and three-day registrations will be available for the Festival. Information on NCC '78 may be obtained from AFIPS Headquarters, 210 Summit Avenue, Montvale NJ 07645, or by calling (201) 391-9810.

Newport Beach CA

A call for papers has been issued for the 1978 Summer Computer Simulation Conference to be held July 24-26, 1978 at the Newport Beach Marriott Hotel. The Conference theme is "Simulation Today." An abstract should be submitted by December 15, 1977 to; Dr. Lance A. Leventhal, Technical Editor, Society for Computer Simulation, PO Box 2228, La Jolla CA 92038.

Long Beach CA

PERCOMP '78 (co-sponsored by the International Computer Society/ SCCS and the Rockwell Hobbyist Computer Club) will be held at the Long Beach Convention Center, Long Beach CA, April 28-30, 1978. PERCOMP is a selling show designed with the home computerist and small-business person in mind. For information concerning seminars, contact James Lindwedel, Technical Program Chairperson, PERCOMP '78, 1833 E. 17th St., Santa Ana CA 92701.

San Francisco CA

COMPCOM SPRING '78 is taking place February 28 to March 2 in San Francisco. For information, contact: COMPCOM SPRING '78, PO Box 639, Silver Spring MD 20901.

NLS LEADS AGAIN!!

THE NEW MS-15 MINISCOPE

PORTABLE
BATTERY OPERATED
15 MEGAHERTZ
OSCILLOSCOPE

FEATURES

- **15 megahertz bandwidth.**
- **External and internal trigger.**
- **Time Base — 0.1 microseconds to 0.5 Sec/div — 21 settings.**
- **Battery or line operation.**
- **Automatic and line sync modes.**
- **Power consumption less than 15W.**
- **Vertical Gain — 0.01 to 50 volts/div — 12 settings.**
- **Weight is only 3 pounds.**

From the originator of the Digital Voltmeter, Non-Linear Systems comes the MS-15 Miniscope. It is a fine electronic instrument with a great deal of measuring capability and excellent accuracy. Its design is modern, utilizing the latest in low-powered integrated circuits, and it is packaged into the smallest practical size. The instrument fits into many briefcases and tool boxes with room to spare.

Operating characteristics have been chosen so that the MS-15 will make all of the measurements needed in servicing most electronic equipment. It is field-portable so its use is not restricted to the bench.

SPECIFICATIONS:

Y Axis
Vertical Input: x1-10mV/div to 10V/div in four ranges, each continuously variable. x2-20mV/div to 20V/div in four ranges, each continuously variable. x5-50mV/div to 50V/div in four ranges, each continuously variable. Accuracy is 3%.
Input Impedance: 1M ohm shunted by 50 pF
Input Voltage: 350 volts peak
Bandwidth (3 db points): DC/DC to 15 MHz; AC 5 Hz to 15 MHz
Internal Calibrator: 1 .0V (±5%) peak-to-peak square wave at 10-20 kHz
Graticule: 4 div Y by 5 div X; 1 div = 0.25 inch. Viewing area: 1.1''H x 1.35''W.

X Axis
Horizontal Input
Bandwidth (3 db points): DC to 200 kHz
Input Impedance: 1M ohm shunted by 50 pF
Sensitivity: 1V/div, 100V peak maximum input.
Time Base: x1, uS-0.1uS/div to 100 uS/div. x 2, uS-0.2uS/div to 200 uS/div. x5, uS-0.5uS/div to 500 uS/div. x1, mS-0.1mS/div to 100 mS/div. x2, mS-0.2mS/div to 200 mS/div. x5, mS-0.5mS/div to 500 mS/div. all in four ranges, each continuously variable. (Range increments are .1, 1, 10, 100.) With vernier in full clockwise position, calibrated time measurements are possible. Accuracy is 3%.

Triggering
Internal: Sweep triggered from internal trigger source.
Automatic: Trigger source is internal calibrator frequency. To be used if there is no other trigger source available to synchronize the sweep.
Line: Trigger is derived from line frequency when using the battery charger.
Slope: Selects sync to positive- or negative-going waveform.
Sensitivity: Less than one division.
External: Controls function as for internal triggering.

Power
Batteries: Three self-contained lead acid ''D'' cells.
Operating Time: Typically 4 hours.
Charging Time Scope Operating: Will run indefinitely but not reach full charge.
Non-operating: Sixteen hours.
External Power: Battery charger 115 vac (220 vac on request), 50-60Hz, less than 15 watts.

Dimensions: 3.1''H x 6.4''W x 8.0''D.
Weight: Three pounds.
Accessories Standard: Tilt stand, battery charger, input cable and miniature banana plugs (4).
Options: Leather carrying case and 10-1 attenuating probe, 10M ohm input.
Warranty: One year parts and labor.

MS-15 with Rechargeable Batteries and Charger
$289.00

Leather Carrying Case

The leather case has 2 separate compartments. One to hold the scope, the other to hold the charger, probe, shoulder strap, etc. The case can be worn on the belt, or over the neck.

The snaps used on the case are "one way", thus accidental striking of the case against an object will not undo the snaps or let it be pulled off your belt.

41-140 **$30.00**

Probe

10 to 1 probe with 10 megohm input. Probe uses spring hook tip for sure connection. Compensation network is located at the connector rather than at the probe, so as to keep size and weight to a minimum.

41-141 **$24.50**

KB

FREE PROBE
Just send or mention this COUPON and MAGAZINE and receive 41-141 PROBE FREE with the purchase of MS-15 MINISCOPE.
Offer good thru Feb. '78

PRIORITY ONE ELECTRONICS
10031 Woodley Ave. Sepulveda CA. 91343

Terms VISA, MC, BAC, check, Money Order, C O D U S Funds Only CA residents add 6% sales tax. Minimum order $10.00 Orders less than $75 00 include 10% shipping and handling, excess refunded. Just in case please include your phone no. Good thru Feb. 15, 1978

phone orders welcome (213) 893-8202

OEM and institutional inquiries invited.

PRIORITY 1 ELECTRONICS a div. of the HEATH Group

Kilobaud Classified

Kilobaud classified advertisements are intended for use by those individuals desiring to buy, sell, or trade used computer equipment or software. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows five lines of about 22 characters each, including spaces and punctuation. The $10 box provides ten lines of type — again, each line is about 22 characters. Minimize capital letters, as they use twice the space of small characters. Payment is required in advance with ad copy. We cannot bill, or accept credit. Oversize ads are not accepted. Each subscriber is limited to two (2) identical ads in any given issue.

Advertising text and payment must reach us 60 days in advance of publication. For example, advertising copy for the March issue (mailed in February) must be in our hands on January 1. The publisher reserves the right to refuse a questionable or not applicable advertisement. Mail advertisements to: KILOBAUD CLASSIFIED, Kilobaud, Peterborough, NH 03458. Do not include any other material with your ad, as it may be delayed.

KSR-33 Teletype: pin-feed, form-feed. No reader or punch; good working condition — $500. George Beamer, Box 417, Christiansburg VA 24073.

For Sale: Line-printer paper, 1000 sets to a box, 2-ply 9-7/8 by 11 inches, $5 per box.
Sprocket-feed paper 8-1/2 by 11 5-ply carbonless, $5 per box, for all Teletypes.
Sprocket-feed paper 8-1/2 by 6 single-ply, for all Teletypes, $5 per box.
I.B.M. Model 731 I/O writer used, in good condition with manual, $350.
NCR digital cassettes 282 feet 800bpi, used, in good condition, $2 each.
Lou Carbaugh, PO Box 398, New Cumberland PA 17070.

6502 Morse code send and receive software. Adaptive speed on rcv — 256 char xmit buffer; use on KIM-1 and other 6502 systems. Assembled source listing and complete documentation, $6. C. H. Galfo, 602 Orange St., Charlottesville VA 22901.

PET-2001 and Radio Shack TRS-80 arrived on campus. I want to survey users and report results to any interested hobbyists. Write: Professor Bill Parks, Walters State Community College, Morristown TN 37814.

TTY ASR-33 with coupler, $750; Mits 4K, $100; 3 P+S, $115; OAE tape reader, $60; Imsai 4K, $110 — assembled, socketed and factory checked. K. Roberts, 10560 Main, #515, Fairfax VA 22030.

For Sale: Viatron terminal with two cassette drives, keyboard, power supply, and RS-232 comm. kit — $300. Also, as above but with one cassette and one 9-track mag. tape drive, $595. Mike Vitale, PO Box 22, Suncook NH 03275 (603) 485-9131 before 6.

Software: North Star Disk Media. Startrek $2, Checker $2, UFO $2, Family Budget $5, Poker & Craps $3, Decision Helper $4. Send for complete list. Software, Box AF, Woodbridge CT 06525.

CORRECTIONS

In my article on the AC-30 control decoder in KB No. 9, pin 21 (reset) of the AY-5-1013 UART should be connected to ground, not left hanging in space.

I must thank a fellow SWTPC owner and fellow Olympian, Mark Zenier, both for catching my error and for not telling me the control decoder didn't work until he had figured out what I did wrong.

Phil Hughes
Olympia WA

Contest!

Bruce A. Artwick is the October winner in our "Best Program of the Month" contest for his article, "3D Computer Graphics," on page 50 of Kilobaud No. 10. Bruce will receive a check for $100; and we are now counting the votes for best program in the November issue.

Winner in the drawing of votes submitted for October is Richard Vilmur of 418 Frederick Ave., Bellwood IL. Mr. Vilmur gets his choice of any book we publish.

After winners have been chosen for November and December, a winner for the year will be selected from among the monthly winners.

Keep voting!

VECTOR-PAK ASSEMBLED MICROCOMPUTER CASES

ADJUSTABLE PACKAGING SYSTEM FOR S-100 BUS MICROCOMPUTERS, COMPATIBLE WITH ALTAIR 8800 and IMSAI 8080 SIZE CARDS.

- Smart looking, deluxe cases unmarred by unsightly screws or fasteners
- Finished in dark blue textured vinyl
- Instantly accessible interiors with slip out covers
- Removable recessed rear and front panels
- Fully adjustable interior mounting systems for any card or card spacing within size limitations. No cutting or drilling necessary
- Perforated bottom cover for cooler operation

PART NO.	DESCRIPTION
VP1 $128.30 NET	Assembled case with perforated bottom cover. Installed mounting struts for card guides and receptacles or mother board. Cards top loaded, spanning front to back. Card guides (12 pair) and chassis plate supplied uninstalled
VP2 $134.30 NET	Same as VP1 except cards oriented side to side

VP2

Vector 8800V

Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16"

1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1

Same as 8800V except plain; less power buses & heat sink.

1-4	5-9	10-24
$14.95	$13.46	$11.96

Vector 3662

3662 6.5" x 4.5" $7.65

3662-2 9.6" x 4.5" $11.45

P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156

3690-12

Card Extender has 100 contacts — 50 per side on .125 centers — Attached connector — is compatible with S-100 Bus Systems $25.00

3690 6.5" 22/44 pin .156 ctrs. Extenders $12.00

Bus P.C. Edge Connectors

R644 P.C. receptacle 22/44 cont. 156 ctrs pierced solder eyelets tails gold $3.45

R644-3 P.C. receptacle 22/44 cont. 156 ctrs wire wrap tails gold $4.49

R681 S-100 DIP solder tails on 140 spaced rows for ALTAIR mother boards $7.35

R681-1 .043 dia. holes gold $7.35

R681-2 .025 sq. 1 wrap lengths posts (.62 long) on .250 spaced rows $5.00

R681-3 pierced solder eyelet tails gold $7.35

NLS

- LM3A 3 dig 1% DC .... $125.
- LM3.5A 3½ dig .5% DC .... $147.
- LM40A 4 dig .1% DC .... $190.
- LM4A 4 dig .03% DC .... $227.

- Rechargeable batteries and charger included
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- No zero adjustment and no full-scale ohms adjust
- Battery-operated — NiCad batteries, also AC line operation
- Large LED display for easy reading without
- Size 2"H x 2.7"W x 4"D
- Parts & Labor guaranteed 1 year
- Tilt stand option $3.50
- Leather case $18.00

.8" LED ALARM CLOCK

12 hr. LED Alarm Clock uses 3½ digit .8" LED Display with AM/PM indicators and colons. Direct drive, PIN to PIN interface with MM5387 I.C. Just add switches, AC Supply, Alarm. Display and I.C. only.

$7.95 or 2/$15.00

8803 MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS

- Kit includes 12 tantalum capacitors for +5, ±12. —12 buses and insulated mounting spacers
- Wiring side shown. Component side bare epoxy glass with white markings for component locations
- G10 epoxy glass board with 2 ounce copper, solder plated and .038 diameter holes for leads
- Solder mask with solder windows on etched circuits to avoid accidental short circuits
- Mounts 11 receptacles with 100 contacts (2 rows) on .125 centers with .250 row spacing. Vector part number R681-2, or mounts 10 receptacles plus interconnections to smaller mother board for expansion
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations
- Large buses +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS) Current ratings are per MIL-STD-275 with 10°C rise
- Fits in Vector-pak enclosures
- Fits in IMSAI 8080 microcomputer as expander board

Price: $29.50

WRAP POST

for .042 dia. holes (all boards on this page)

T-44 pkg 100 .... $2.28

T-44 pkg 1000 .... $14.00

A-13 hand installing tool $2.80

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

APPLIED MARKETING

- For Auto, Home, Office
- Small in size (2x2¼x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE:
- LCD-101, portable model runs on self-contained batteries for better than a year.
- LCD-102, runs on 12 volt system and is back lighted.
- LCD-101 or LCD-102 your choice .... $34.95 ea.
- Clear desk stand for $2.00

SLIT-N-WRAP

Wraps insulated wire on .025" square posts

FOUR TIMES FASTER than regular manual wrap-post tools

Vector

NO PRE-STRIPPING* NO PRE-CUTTING* SPOOL-FED WIRE*

*The spooled wire passes through the tool past a slitting edge next to the wrap post. A narrow longitudinal cut is made in the insulation where it presses the square post corner, the bared copper is indented by the sharp edge (7 turns = 28 contacts). Insulation is slit where wrapped but not between terminal posts. Wire fed straight out of the tool.

P180 with two 100' spools of 28 ga. wire $24.95

P160-4T charger, wire includes $75.00

"Daisy-chain" runs reliable and easy to use Manual or power operation

Buy P180 and get 1 package of wire for 1c

Buy P160-4T and get 3 packages of wire for 1c

SLIT-N-WRAP WIRE NO. 28 GAGE INSULATED WIRE: 100' SPOOLS

$2.75 Package

W28-2-A Pkg. 3 Green — W28-2-C Pkg. 3 Clear

W28-2-B Pkg. 3 Red — W28-2-D Pkg. 3 Blue

ACE • All-Circuit Evaluator for fast, solderless, plug-in circuit building and testing

Just plug in any components with leads to .032" dia. Interconnect with solid wire up to 20 ga. Assembled models too!

ACE Model No.	Tie Points	DIP Capacity	No. Buses	No. Posts	Board Size (inches)	Price Each
200-K (kit)	728	8 (16's)	2	2	4-9/16 x 5-9/16	$18.95
208 (assem)	872	8 (16's)	8	2	4-9/16 x 5-9/16	29.95
201-K (kit)	1032	12 (14's)	2	2	4-9/16 x 7	24.95
212 (assem)	1224	12 (14's)	8	2	4-9/16 x 7	34.95
218 (assem)	1760	18 (14's)	10	2	6½ x 7-1/8	46.95
227 (assem)	2712	27 (14's)	28	4	8 x 9¼	59.95
236 (assem)	3648	36 (14's)	36	4	10-1/4 x 9-1/4	79.95

Gold-anodized aluminum base ground, non-corrosive nickel-silver terminals, 4 rubber feet

Model 236

16 K EPROM

TMS 2716

PRIORITY ONE ELECTRONICS is first to bring you FACTORY PRIME T.I. TMS-2716JL 16K EPROM programs like a 2708.

$39.95

MICRO-KLIP

for .042 dia. holes (all boards on this page)

T42-1 pkg. 100 .... $1.50

T42-1 pkg. 1000 .... $11.00

P-149 hand installing tool $2.03

NLS

SC-5 With Rechargeable Batteries & Charger Unit $89

FM-7 With Rechargeable Batteries & Charger Unit $195

Features include: • By using the new NLS SC-5 Prescaler, the range of the FM-7 Frequency Meter, which is 10 Hz to 60 MHz, may be extended to 512 MHz (the upper VHF & UHF frequency bands) • The FM-7 utilizes an LED readout, providing 7-digit resolution • The FM-7 can be calibrated to an accuracy of 0.00001% • The SC-5 is accurate to one part per million • Each unit has 30 millivolts sensitivity, is battery powered and has a charger unit included • Dimensions of each are 1.9" W x 2.2" W x 3.9" D • The units may be obtained separately or as a "Frequency Duo" • Parts & Labor guaranteed 1 year

Tilt stand option $3.50

Leather case $18.00

2708 8K 450 ns EPROM FACTORY PRIME

1-9	$14.75
10-24	$13.50
25-99	$12.50
100+	$11.95

24 PIN DIP PLUGS WITH COVERS

3 / $1.00

40 / $10.00

1/16 Vector BOARD

.042 dia holes on 0.1 spacing for IC's

Phenolic PART NO.	SIZE	PRICE
64P44-062XXP	4.5 x 6.5"	$1.49
169P44-062XXP	4.5 x 17"	$3.51

Epoxy Glass PART NO.	SIZE	PRICE
64P44-062	4.5 x 6.5"	$1.70
84P44-062	4.5 x 8.5"	$2.10
169P44-062	4.5 x 17"	$4.30
169P84-062	8.5 x 17"	$6.39

100 K Joypot

$4.95 Buy 3 Get 1 FREE

3 LEVEL GOLD WIRE WRAP SOCKETS

	1-24	25-49	50-99
8 pin	.47	.42	.37
10 pin	.45	.41	.37
14 pin	.39	.38	.37
16 pin	.43	.42	.40
18 pin	.75	.68	.62
22 pin	1.00	.97	.95
24 pin	1.00	.94	.75
28 pin	1.09	.98	.84
36 pin	1.59	1.45	1.30
40 pin	1.49	1.39	1.29

SPECIAL

14CS2 100 for $14.00

16CS2 100 for $16.00

14 pin CS2 10 for $1.60

16 pin CS2 8 for $1.60

These low cost DIP sockets will accept both standard width chips, plugs and chips for use with chips. The plugs offer a low profile height of only .130" above the board. These sockets are end stackable.

Price Breakthrough! MA1003 CAR CLOCK

Bright Green Fluorescent Display Crystal Time Base Assembled, just add switches and 12 VDC.

$17.95

Solitron DEVICES INC CMOS

TAKE ADVANTAGE OF THIS SPECIAL PURCHASE OF 100% FACTORY PRIME PARTS

CM4006AE	2.00
CM4007AE	8 / 1.00
CM4008AE	3 / 1.00
CM4011AE	8 / 1.00
CM4012AE	5 / 1.00
CM4013AE	4 / 1.00
CM4014AE	3 / 2.00
CM4015AE	3 / 2.00
CM4017AE	2 / 1.00
CM4018AE	1.00
CM4019AE	4 / 1.00
CM4020AE	3 / 2.00
CM4022AE	1.00
CM4023AE	8 / 1.00
CM4024AE	3 / 1.00
CM4025AE	5 / 1.00
CM4030AE	7 / 1.00
CM4035AE	3 / 2.00
CM4037AE	2.00
CM4041AE	2 / 1.00
CM4047AE	1.00
CM4049AE	4 / 1.00
CM4050AE	4 / 1.00
CM4116AE	4 / 1.00

(10% Discount on 100 pcs. or More, if a Minimum 25 per Part No. are Purchased.)

IC TEST CLIPS

MODEL	Row To Row Dimension	PART NUMBER	PRICE
TC 8	3 IN	923695	4.50
TC 14	3 IN	923698	4.50
TC 16	3 IN	923700	4.75
TC 16 LSI	5/6 IN	923702	8.95
TC 18	3 IN	923703	10.00
TC 20	3 IN	923704	11.00
TC 22	4 IN	923706	11.50
TC 24	5/6 IN	923714	13.85
TC 28	5/6 IN	923718	15.25
TC 36	5/6 IN	923720	19.95
TC 40	5/6 IN	923722	21.00

3682

3682 9.6" x 4.5" $10.97

3682-2 6.5" x 4.5" $9.81

Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156

3677

3677 9.6" x 4.5" $10.90

3677-2 6.5" x 4.5" $9.74

Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156

PRIORITY ONE ELECTRONICS

10031 Woodley Ave. Sepulveda CA. 91343

Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case please include your phone no. Good thru Feb. 15, 1978

Send for our latest brochure. **phone orders welcome (213) 893-8202** OEM and Institutional inquiries invited.

BREAD BOARD JUMPER WIRE KIT

Each kit contains 350 wires cut to 14 different lengths from 0.1" to 5.0'

Each wire is stripped and the leads are bent 90° for easy insertion

Wire length is classified by color coding

All wire is solid tinned 22 gauge with PVC insulation.

The wires come packed in a convenient plastic box.

JK1 . . . .$10.00 / kit

SOCKET JUMPERS

Mates with two rows of .025" sq. or dia. posts on patterns of .100" centers and shielded receptacles. Probe access holes in back. Choice of 6" or 18" length.

Part No.	No. of Contacts	Length	Price
924003-18R	26	18"	$ 5.38 ea.
924003-06R	26	6"	4.78 ea.
924005-18R	40	18"	8.27 ea.
924005-06R	40	6"	7.33 ea.
924006-18R	50	18"	10.31 ea.
924006-06R	50	6"	9.15 ea.

JUMPER HEADERS

Solder to PC boards for instant plug-in access via socket-connector jumpers. .025" sq. posts. Choice of straight or right angle.

Part No.	No. of Posts	Angle	Price
923863-R	26	straight	$1.28 ea.
923873-R	26	right angle	1.52 ea.
923865-R	40	straight	1.94 ea.
923875-R	40	right angle	2.30 ea.
923866-R	50	straight	2.36 ea.
923876-R	50	right angle	2.82 ea.

INTRA-CONNECTOR

Provides both straight and right angle functions. Mates with standard .10" x .10" dual row connectors (i.e. 3m, Ainsley, etc.) Permits quick testing of inaccessible lines.

Part No.: 922576-26 **No. of contacts:** 26 **Price** $6.90 ea.

INTRA-SWITCH

Permits instant line-by-line switching for diagnostic or QA testing. Switches actuated with pencil or probe tip. Mates with standard .10" x .10" dual-row connectors. Low profile design. Switch buttons recessed to eliminate accidental switching.

Part No.: IS-26 **No. of contacts:** 26 **Price** $13.80 ea.

CRYSTALS

THESE FREQUENCIES ONLY

Part #	Frequency	Case/Style	Price
CY1A	1.000 MHz	HC33/U	$5.95
CY2A	2.000 MHz	HC33/U	$5.95
CY2.01	2.010 MHz	HC33/U	$1.95
CY3A	4.000 MHz	HC18/U	$4.95
CY7A	5.000 MHz	HC18/U	$4.95
CY12A	10.000 MHz	HC18/U	$4.95
CY14A	14.31818 MHz	HC18U	$4.95
CY19A	18.000 MHz	HC18/U	$4.95
CY22A	20.000 MHz	HC18/U	$4.95
CY30B	32.000 MHz	HC18/U	$4.95

CONNECTORS

PRINTED CIRCUIT EDGE-CARD

156 Spacing-Tin-Double Read-Out

Bifurcated Contacts — Fits .054 to .070 P.C. Cards

15/30	PINS (Solder Eyelet)	$1.95
18/36	PINS (Solder Eyelet)	$2.49
22/44	PINS (Solder Eyelet)	$2.95
50/100	PINS (Wire Wrap)	$6.95
50/100A (.100 Spacing)	PINS (Wire Wrap)	$6.95

25 PIN-D SUBMINATURE (RS232)

DB25P	PLUG	$3.25
DB25S	SOCKET	$4.95

205-CB HEAT SINKS 680-.75A

205-CB	Beryllium Copper w-black finish for T0-5	$ .25
291-.36H	Aluminum for T0-220 Transistors & Regulators	$ .25
680-.75A	Black Anodized Aluminum for T0-3	$1.60
Dude 4	Black Anodized Aluminum — predrilled mounting holes for T0-3 — 4¾ x 1¼ x 2"	$1.75

Etching Kits

(cannot be shipped via air)

32 X A-1	P C Etch Materials Kit enough for 5 circuit boards	$29.95 ea.
27 X A-1	Etched Circuit Kit Complete kit — only add water	$ 9.95 ea.

Plugboards

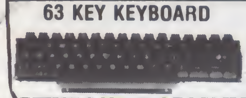
3662	6 5 X 4 5 X 1/16 Epoxy glass P-Pattern-44 P C Tabs-spaced 156	$ 6.95 ea.
22/44	Mating connector for plugboard — 22 pin double readout	$ 2.95 ea.
8800V	Universal Microcomputer/Processor plugboard — Epoxy Glass — complete with heatsink and mounting hardware 5 313 X 10 X 1/16 copper clad	$19.95 ea.

Vector

1/16 VECTOR BOARD

	0.1 Hole Spacing Part No	P-Pattern L	W	Price 1-9	10 up
PHENOLIC	64P44 062XXXP	4 50	6 50	1 72	1.54
	169P44 062XXXP	4 50	17.00	3 69	3.32
EPOXY GLASS	64P44 062WE	4 50	6 50	2 07	1.86
	84P44 062WE	4.50	8.50	2.56	2 31
	169P44 062WE	4 50	17.00	5.04	4 53
	169P84 062WE	8 50	17.00	9 23	8 26
EPOXY GLASS COPPER CLAD	169P44 062WEC1	4 50	17.00	6 80	6 12

63 KEY KEYBOARD $29.95



This keyboard features 63 unencoded SPST keys unattached to any kind of P C B A very solid molded plastic 13 x 4 base suits most applications

HD0165	Encoder Chip (encodes 16 Keys)	$7.95 ea.
AY-5-2376	Encoder Chip (encodes 88 Keys)	$14.95 ea.

MICROPROCESSOR COMPONENTS

8080A	CPU	$16.00	CDP1802	CPU	$19.95
8212	8 Bit Input/Output	4.95	MC6800	8 Bit MPU	24.95
8214	Priority Interrupt Control	15.95	MC6820	Periph. Interface Adapter	15.00
8216	Bi-Directional Bus Driver	6.95	MC6810AP1	128 x 8 Static RAM	6.00
8224	Clock Generator/Driver	9.95	MC6830L8	1024 x 8 Bit ROM	15.00
8228	System Controller Bus Driver	10.95	Z80	CPU	29.95

CPU'S

8080A	Super 8008	16.00
2650	8 BIT MPU	26.50
P8085	CPU	29.95

SR'S

2504	1024 Dynamic	$ 3.95
2518	Hex 32 BIT	7.00
2519	Hex 40 BIT	4.00
2522	Dual 132 Bit SSR	2.95
2524	512 Dynamic	1.95
2525	1024 Dynamic	3.00
2527	Dual 256 BIT	3.95
2529	Dual 512 BIT	4.00
2532	Quad 80 BIT	3.95
2533	1024 Static	5.95
3341	Fifo	6.95
74LS670	16 x 4 Reg	3.95

UART'S

AY-5-1013	30K Baud	$5.95

ROM'S

2513(2140)	Char. Gen.-upper case	$ 9.95
2513(3021)	Char. Gen.-lower case	9.95
2516	Char. Gen.	10.95
MM5230	2048 BIT (512 x 4 on 256 x 8)	1.95

RAM'S

1101	256 x 1	Static	$ 1.49
2101	256 x 4	Static	5.95
2102	1024 x 1	Static	1.75
2107/5280	4096 x 1	Dynamic	4.95
2111	256 x 4	Static	6.95
7489	16 x 4	Static	2.49
8101	256 x 4	Static	5.95
8111	256 x 4	Static	6.95
8599	16 x 4	Static	3.49
21L02/91L02	1024 x 1	Static	2.25
74200	256 x 1	Static	6.95
93421	256 x 1	Static	2.95
MM5262	2K x 1	Dynamic	2 for 1.00
UPD414(2104)4K		Dynamic 16 Pin	5.95

PROMS

1702A	2048	Famos	$ 9.95
5203	2048	Famos	14.95
82S23	32 x 8	Open C	5.00
82S123	32 x 8	Tristate	5.00
74S287	1024	Static	7.95
3601	256 x 4	Fast	3.95
2708	8K	Eprom	19.95
2716	16K	Eorom	34.95
6301-1	1024	Tri-State Bipolar	3.49
6330-1	256	Open Collector Bipolar	2.95

SPECIAL REQUESTED ITEMS

FCM3817	$5.00	11C90	19.95	7205	19.95	9368	3.95
AY-3-8500-1	8.95	4N33	3.95	ICM7045	24.95	LD110/111	25.00/set
AY-5-9100	17.50	8T20	7.50	ICM7207	7.50	95H90	11.95
AY-5-9200	14.95	8T97	2.00	ICM7208	22.00	MC3061P	3.50
AY-5-9500	4.95	HD0165	7.95	ICM7209	7.50	MC4016 (74416)	7.50
AY-5-2376	14.95	MCM6571	13.50	MK50240	17.50	MC1408L7	8.95
9374	1.95	MCM6574	13.50	DS0026CH	3.75	MC1408L8	9.95
82S115	25.00	MCM6575	13.50	TIL308	10.50		

CLOCK CHIPS

MM5309	$9.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5841	9.95
CT7001	5.95

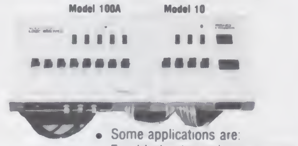
PARATRONICS

Featured on February's Front Cover of Popular Electronics

Logic Analyzer Kit Model 100A

$229.00/kit

Model 100A assembled $295.00



- Analyzes any type of digital system
- Checks data rates in excess of 8 million words per second
- Trouble shoot TTL, CMOS, DTL, RTL, Schottky and MOS families
- Displays 16 logic states up to 8 digits wide
- See ones and zeros displayed on your CRT, octal or hexadecimal format
- Tests circuits under actual operating conditions
- Easy to assemble — comes with step-by-step construction manual which includes 80 pages on logic analyzer operation.

(Model 100A Manual - $4.95)

- Some applications are:
  - — Troubleshooting microprocessor address, instruction, and data flow
  - — Examine contents of ROMS
  - — Tracing operation of control logic
  - — Checking counter and shift register operation
  - — Monitoring I/O sequences
  - — Verifying proper system operations during testing

PARATRONICS TRIGGER EXPANDER - Model 10

Adds 16 additional bits. Provides digital delay and qualification of input clock and 24-bit trigger word. — Connects direct to Model 100A for integrated unit).

Model 10 Kit — $229.00
Model 10 Assembled — $295.00
Baseplate — $9.95
Model 10 Manual — $4.95

B+K PRECISION

3½-Digit Portable DMM

- Overload Protected
- 3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv, 1Va, 0 1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage ±0-1000V/ AC Voltage 0-1000V
Freq Response 50-400 HZ
DC/AC Current: 0-100mA
Resistance 0-10 meg ohm
Size 6 4" x 4 4" x 2"

Model 2800 $99.95

Comes with test leads, operating manual and spare fuse

Accessories:

AC Adapter BC-28	$9.00
Rechargeable Batteries BP-26	20.00
Carrying Case LC-28	7.50

CSC

100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95

ACCESSORIES FOR MAX 100:

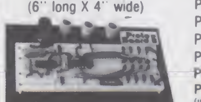
Mobile Charger Eliminator use power from car battery — **Model 100 — CLA $3.95**

Charger/Eliminator use 110 V AC — **Model 100 — CAI $9.95**

CONTINENTAL SPECIALTIES

PROTO BOARD 6

$15.95
(6" long X 4" wide)



Other CS Proto Boards

PB100 - 4.5" x 6"	$ 19.95
PB101 - 5.8" x 4.5"	29.95
PB102 - 7" x 4.5"	39.95
PB103 - 9" x 6"	59.95
PB104 - 9.5" x 8"	79.95
PB203 - 9.75 x 6½ x 2¾	80.00
PB203A - 9.75 x 6½ x 2¾ (includes power supply)	129.95

PROTO CLIPS

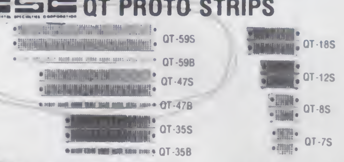
14 PIN	$4.50
16 PIN	4.75
24 PIN	8.50
40 PIN	13.75

DESIGN MATES

DM1 - Circuit Designer $69.95
DM2 - Function Generator $74.95
DM3 - RC Bridge $74.95

LOGIC MONITOR for DTL, HTL, TTL or CMOS Devices **$84.95**

CSC QT PROTO STRIPS



QT type	# holes	price
QT-59S	590	12 50
QT-59B	bus strip	2 50
QT-47S	470	10 00
QT-47B	bus strip	2 25
QT-35S	350	8 50
QT-35B	bus strip	2 00
QT-18S	180	4 75
QT-12S	120	3 75
QT-8S	80	3 25
QT-7S	70	3 00

Experimentor 300	$ 9.95
Experimentor 600	$10.95

$5.00 Minimum Order — U.S. Funds Only
California Residents — Add 6% Sales Tax

Spec Sheets - 25¢ — Send 35¢ Stamp for 1978 Catalog
Dealer Information Available

NOW OPEN SATURDAYS

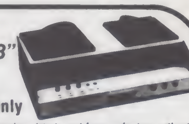
James ELECTRONICS

1978 CATALOG NOW AVAILABLE

1021-A HOWARD AVE., SAN CARLOS, CA. 94070
PHONE ORDERS WELCOME — (415) 592-8097
All Advertised Prices Good Thru January

J1

The Incredible "Pennywhistle 103"



$129.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications for the deaf. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method	Frequency-Shift Keying, full-duplex (half-duplex selectable)
Maximum Data Rate	300 Baud
Data Format	Asynchronous Serial (return to mark level required between each character)
Receive Channel Frequencies	2025 Hz for space, 2225 Hz for mark
Transmit Channel Frequencies	Switch selectable: Low (normal) = 1070 space, 1270 mark. High = 025 space, 2225 mark
Receive Sensitivity	−46 dbm accoustically coupled.
Transmit Level	−15 dbm nominal. Adjustable from −6 dbm to −20 dbm
Receive Frequency Tolerance	Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface	EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements	120 VAC, single phase, 10 Watts
Physical	All components mount on a single 5" by 9" printed circuit board. All components included

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscilloscope to align

ERC AM/FM 8-Track Stereo Receiver With BSR Changer

- PLL System
- BSR Record Changer
- Slide Controls
- Automatic AFC Control
- 4 Speaker Output
- Walnut finish vinyl covered wood veneer with smoke dust cover

Size: 20"W x 9⅜"H x 15¼"D

Model 8365 $149.95

DIGITAL STOPWATCH

- Bright 6 Digit LED Display
- Times to 59 minutes 59 59 seconds
- Crystal Controlled Time Base
- Three Stopwatches in One
  Times Single Event — Split & Taylor
- Size 4 5" x 2 15" x .90" (4½ ounces)
- Uses 3 Penlite Cells

Kit — $39.95
Assembled — $49.95
Heavy Duty Carry Case $5.95

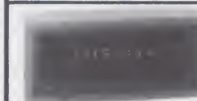
Stop Watch Chip Only (7205) $19.95

IMC 3½ DIGIT DPM KIT

- New Bipolar Unit
- Auto Zeroing
- .5" LED
- Auto Polarity
- Low Power
- Single IC Unit

Model KB500 DPM Kit	**$49.00**
Model 311D-5C-5V Power Kit	**$17.50**

JE700 CLOCK



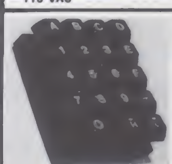
The JE700 is a low cost digital clock but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6 x 2½ x 1. It utilizes a MAN72 high brightness readout and the MM5314 clock chip

12 or 24 Hour

115 VAC

KIT ONLY **$16.95**

HEXADECIMAL ENCODER 19-KEY PAD



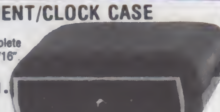
- 1 - 0
- ABCDEF
- Shift Key
- *2 Optional Keys*

$10.95 each

INSTRUMENT/CLOCK CASE

Injection molded unit. Complete with red bezel. 4½" x 4" x 1-9/16".

$3.95 ea.



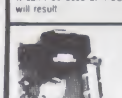
JE803 PROBE

The Logic Probe is a unit which is for the most part indespensible in trouble shooting logic families TTL DTL RTL CMOS. It derives the power it needs to operate directly off of the circuit under test drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols (HI) 1 (LOW) 0 (PULSE) P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result

$9.95 Per Kit

printed circuit board

T²L 5V 1A Supply



This is a standard TTL power supply using the well known LM309K regulator IC to provide a solid 1 AMP of current at 5 volts. We try to make things easy for you by providing everything you need in one package including the hardware for only

$9.95 Per Kit

7400N TTL

SN7400N	.16	SN7472N	.39	SN74160N	1.25
SN7401N	.18	SN7473N	.39	SN74161N	.99
SN7402N	.20	SN7474N	.35	SN74162N	1.95
SN7403N	.20	SN7475N	.50	SN74163N	.99
SN7404N	.20	SN7476N	.35	SN74164N	.99
SN7405N	.20	SN7479N	5.00	SN74165N	.99
SN7406N	.35	SN7480N	.50	SN74166N	1.25
SN7407N	.35	SN7482N	.99	SN74167N	3.25
SN7408N	.20	SN7483N	.70	SN74170N	2.10
SN7409N	.25	SN7485N	.89	SN74172N	6.00
SN7410N	.20	SN7486N	.39	SN74173N	1.50
SN7411N	.30	SN7488N	3.50	SN74174N	1.25
SN7412N	.35	SN7489N	2.49	SN74175N	.99
SN7413N	.69	SN7490N	.50	SN74176N	.79
SN7414N	.70	SN7491N	.75	SN74177N	.79
SN7416N	.35	SN7492N	.49	SN74179N	2.49
SN7417N	.35	SN7493N	.49	SN74180N	.99
SN7420N	.20	SN7494N	.79	SN74181N	2.49
SN7421N	.39	SN7495N	.79	SN74182N	.95
SN7422N	.49	SN7496N	.79	SN74184N	1.95
SN7423N	.37	SN7497N	3.00	SN74185N	1.95
SN7425N	.29	SN74100N	1.25	SN74186N	15.00
SN7426N	.29	SN74107N	.39	SN74187N	8.00
SN7427N	.37	SN74109N	.95	SN74188N	3.95
SN7429N	.42	SN74116N	1.95	SN74190N	1.79
SN7430N	.25	SN74121N	.39	SN74191N	1.25
SN7432N	.35	SN74122N	.39	SN74192N	.89
SN7437N	.35	SN74123N	.50	SN74193N	.89
SN7438N	.35	SN74125N	.60	SN74194N	1.25
SN7439N	.25	SN74126N	.60	SN74195N	.75
SN7440N	.21	SN74132N	1.25	SN74196N	1.00
SN7441N	.89	SN74136N	.95	SN74197N	1.00
SN7442N	.69	SN74141N	1.15	SN74198N	1.75
SN7443N	.75	SN74142N	2.95	SN74199N	1.75
SN7444N	.75	SN74143N	3.25	SN74200N	5.59
SN7445N	.75	SN74144N	3.00	SN74251N	1.79
SN7446N	.89	SN74145N	1.15	SN74279N	.90
SN7447N	.69	SN74147N	2.35	SN74283N	2.25
SN7448N	.89	SN74148N	2.00	SN74284N	6.00
SN7450N	.25	SN74150N	1.25	SN74285N	6.00
SN7451N	.25	SN74151N	.79	SN74365N	.75
SN7453N	.25	SN74153N	.89	SN74366N	.75
SN7454N	.25	SN74154N	1.25	SN74367N	.75
SN7459A	.25	SN74155N	.89	SN74368N	.75
SN7460N	.25	SN74156N	.89	SN74390N	2.25
SN7470N	.45	SN74157N	.89	SN74393N	2.25

20% Discount for 100 Combined 7400's

CMOS

CD4000	.23	CD4044	.89	CD4520	1.29
CD4001	.23	CD4046	1.79	MC14562	14.50
CD4002	.23	CD4047	2.50	CD4566	2.25
CD4006	1.19	CD4048	1.35	MC14583	3.50
CD4007	.25	CD4049	.49	**74C00 Series**	
CD4009	.49	CD4050	.49	74C00	.39
CD4010	.49	CD4051	1.19	74C02	.55
CD4011	.23	CD4053	1.19	74C04	.75
CD4012	.25	CD4056	1.49	74C10	.65
CD4013	.39	CD4059	9.95	74C14	3.00
CD4014	1.39	CD4060	1.49	74C20	.65
CD4015	1.19	CD4066	.79	74C30	.65
CD4016	.49	CD4068	.39	74C42	2.15
CD4017	1.19	CD4069	.45	74C73	1.50
CD4018	.99	CD4070	.55	74C74	1.15
CD4019	.49	CD4071	.23	74C89	4.00
CD4020	1.19	CD4072	.49	74C90	3.00
CD4021	1.39	CD4076	1.39	74C93	2.00
CD4022	1.19	CD4081	.23	74C95	2.00
CD4023	.23	CD4082	.23	74C107	1.25
CD4024	.79	CD4098	2.49	74C151	2.90
CD4025	.23	MC14409	14.95	74C154	3.00
CD4026	2.25	MC14410	14.95	74C157	2.15
CD4027	.69	MC14411	14.95	74C160	3.25
CD4028	.89	MC14419	4.95	74C161	3.25
CD4029	1.19	MC14506	.75	74C163	3.00
CD4030	.49	MC14507	.99	74C164	3.25
CD4035	.99	CD4508	3.95	74C173	2.60
CD4040	1.19	CD4510	1.39	74C193	2.75
CD4041	1.25	CD4511	1.29	74C195	2.75
CD4042	.99	CD4515	2.95	80C95	1.50
CD4043	.89	CD4518	1.29	80C97	1.50

LINEAR

LM300H	.80			LM739N	1.19
LM301H	.35	LM340T-8	1.25	LM741CH	.35
LM301CN	.35	LM340T-12	1.25	LM741CN	.35
LM302H	.75	LM340T-15	1.25	LM741-14N	.39
LM304H	1.00	LM340T-18	1.25	LM747H	.79
LM305H	.60	LM340T-24	1.25	LM747N	.79
LM307CN	.35	LM350N	1.00	LM748H	.39
LM308H	1.00	LM351CN	.65	LM748N	.39
LM308CN	1.00	78MG	1.75	LM1303N	.90
LM309H	1.10	LM370N	1.15	LM1304N	1.19
LM309K	1.25	LM373N	3.25	LM1305N	1.40
LM310CN	1.15	LM377N	4.00	LM1307N	.65
LM311H	.90	LM380N	1.25	LM1310N	2.95
LM311N	.90	LM380CN	.99	LM1351N	1.65
LM317K	7.50	LM381N	1.79	LM1414N	1.75
LM318CN	1.50	LM382N	1.79	LM1458CN	.59
LM319N	1.30	NE501K	8.00	LM1496N	.95
LM320K-5	1.35	NE510A	6.00	LM1556V	1.75
LM320K-5.2	1.35	NE529A	4.95	LM2111N	1.95
LM320K-12	1.35	NE531H	3.00	LM2901N	2.95
LM320K-15	1.35	NE536T	6.00	LM3053	1.50
LM320T-5	1.25	NE540L	6.00	LM3065N	.69
LM320T-5.2	1.25	NE550N	1.30	LM3900N(3401)	.49
LM320T-8	1.25	NE555V	.39	LM3905N	.89
LM320T-12	1.25	NE560B	5.00	LM3909	1.25
LM320T-15	1.25	NE561B	5.00	LM5556N	1.85
LM320T-18	1.25	NE562B	5.00	MC5558V	1.00
LM320T-24	1.25	NE565H	1.75	LM7525N	.90
LM323K-5	5.95	NE565N	1.25	LM7534N	.75
LM324N	1.80	NE566CN	1.75	8038B	4.95
LM339N	.99	NE567H	1.95	LM75450	.49
LM340K-5	1.35	NE567V	1.49	75451CN	.39
LM340K-6	1.35	LM703CN	.45	75452CN	.39
LM340K-8	1.35	LM709H	.29	75453CN	.39
LM340K-12	1.35	LM709N	.29	75454CN	.39
LM340K-15	1.35	LM710N	.79	75491CN	.79
LM340K-18	1.35	LM711N	.39	75492CN	.89
LM340K-24	1.35	LM723H	.55	75494CN	.89
LM340T-5	1.25	LM723N	.55	RC4194	5.95
LM340T-6	1.25	LM733N	1.00	RC4195	3.25

74LS00 TTL

74LS00	.29			74LS155	1.25
74LS02	.29			74LS157	1.50
74LS03	.29	74LS75	.69	74LS160	1.95
74LS04	.35	74LS78	.49	74LS161	1.95
74LS05	.35	74LS83	1.75	74LS162	1.95
74LS08	.29	74LS85	2.49	74LS163	1.95
74LS10	.29	74LS86	.49	74LS164	1.95
74LS13	.69	74LS90	.89	74LS175	1.95
74LS14	1.75	74LS92	.89	74LS181	3.69
74LS20	.29	74LS93	.89	74LS190	2.49
74LS26	.39	74LS95	1.50	74LS191	2.49
74LS27	.39	74LS96	1.89	74LS192	2.49
74LS28	.39	74LS107	.59	74LS193	2.49
74LS30	.29	74LS109	.59	74LS194	1.89
74LS32	.39	74LS112	.59	74LS195	1.89
74LS40	.39	74LS123	1.25	74LS253	1.75
74LS42	1.25	74LS132	1.25	74LS257	1.75
74LS47	1.25	74LS136	.59	74LS260	.55
74LS51	.29	74LS138	1.25	74LS279	.79
74LS55	.29	74LS139	1.25	74LS367	.99
74LS73	.49	74LS151	1.25	74LS368	.99
74LS74	.49	74LS153	1.25	74LS670	3.95

BUGBOOK®

Continuing Education Series

BUGBOOKS I and II — **$17.00 per set**
by Peter R. Rony, David G. Larsen, WB4HYJ
Sold as a set these two books outline over 90 experiments designed to teach the reader all he will need to know about TTL logic chips to use them in conjunction with microprocessor systems. You'll learn about the basic concepts of digital electronics including gates, flip-flops, latches, buses, decoders, multiplexers, demultiplexers, LED displays, RAM's, ROM's, and much, much more

BUGBOOK IIa — **$5.00**
by Peter R. Rony, David G. Larsen, WB4HYJ
This volume will introduce you to the fabulous UART chip — that all important interface between data terminals, etc., and your microcomputer. It also covers current loops, and the RS 232C interface standard. Particularly recommended for any RTTY enthusiast.

THE 555 TIMER APPLICATIONS SOURCEBOOK WITH EXPERIMENTS — **$6.95**
by Howard M. Berlin
This book shows you what the 555 timer is and how to use it. Included are over 100 various design techniques, equations and graphs to create "ready-to-go" timers, generators, power supplies, measurement and control circuits, party games, circuits for the home and automobile, photography, music and Amateur Radio

BUGBOOK III — **$15.00**
by Peter R. Rony, David G. Larsen, WB4HYJ, Jonathan A. Titus
Here is the book that puts it all together. Besides having much valuable text there are a series of experiments in which the reader completely explores the 8080 chip pin by pin and introduces you to the Mark 80 microcomputer, a unique easily interfaced system. It is recommended that you have the background on the BUGBOOKS I & II before proceeding with BUGBOOK III

INSTRUCTOR'S MANUAL — **$3.00**
Necessary for instruction of Bugbook I and II. Answers questions regarding experiments, suggestions for further reading, philosophy of authors approach to digital electronics. A must for self-teaching individuals

BUGBOOK V and VI — **$19.00 per set**
by David G. Larsen, Peter R. Rony, Jonathan A. Titus
Experiments in digital electronics, 8080A microcomputer programming and 8080A microcomputer interfacing. An integrated approach to self-instructed basic digital electronics, breadboarding and 8080A interfacing/programming. Bugbook VI integrates the digital concepts of Bugbook V into a treatment of 8080A microcomputer programming and interfacing. Detail & laboratory experiments included with each book

OP AMP MANUAL — **$9.00**
An experiment guide to application of operational amplifiers. Over 25 experiments on all phases of Op Amps

DBUG — **$5.00**
8080 interpretive debugger. A program for entering, debugging and storing assembly language programs

CMOS-M — DESIGNERS PRIMER AND HANDBOOK — **$6.00**
Starts at basic structure of CMOS devices through integration into MSI

COMPLETE MANUAL FOR DIGITAL CLOCKS by John Weiss and John Brooks
Familiarizes technician or hobbyist with basic theories behind digital clocks. Includes trouble shooting guides, basic characteristics of clocks, soldering techniques, clock component data sheets and construction tips — **$3.95**

DISCRETE LEDS

.125" dia.			.190" dia.		
XC209	Red	5/$1	XC111	Red	10/$1
XC209	Green	4/$1	XC111	Green	4/$1
XC209	Orange	4/$1	XC111	Yellow	4/$1
XC209	Yellow	4/$1	XC111	Orange	4/$1

.200" dia			.185" dia.			.200" dia.		
XC22	Red	5/$1	XC526	Red	5/$1	XC556	Red	5/$1
XC22	Green	4/$1	XC526	Green	4/$1	XC556	Green	4/$1
XC22	Yellow	4/$1	XC526	Yellow	4/$1	XC556	Yellow	4/$1
XC22	Orange	4/$1	XC526	Clear	4/$1	XC556	Orange	4/$1
SSL-22	RT	4/$1				XC556	Clear	7/$1

.085" dia. MV50 - Red 6/$1

INFRA-RED LED ¼" x ¼" x 1/16" Flat **5/$1.00**

DISPLAY LEDS

TYPE	POLARITY	HT	PRICE
MAN 1	Common Anode-red	.270	2.95
MAN 2	5 x 7 Dot Matrix-red	.300	4.95
MAN 3	Common Cathode-red	.125	.39
MAN 4	Common Cathode-red	.187	1.95
MAN 7	Common Anode-red	.270	1.25
MAN 7G	Common Anode-green	.270	1.95
MAN 7Y	Common Anode-yellow	.270	1.95
MAN 52	Common Anode-green	.300	1.00
MAN 71	Common Anode-red	.300	1.25
MAN 72	Common Anode-red	.300	1.25
MAN 74	Common Cathode-red	.300	1.50
MAN 81	Common Anode-yellow	.300	1.00
MAN 82	Common Anode-yellow	.300	1.00
MAN 84	Common Cathode-yellow	.300	1.00
MAN 3620	Common Anode-orange	.300	1.00
MAN 3630	Common Anode-orange ±1	.300	1.35
MAN 3640	Common Cathode-orange	.300	1.75
MAN 4610	Common Anode-orange	.300	1.00
MAN 4640	Common Cathode-orange	.400	1.00
MAN 4710	Common Anode-red ±1	.400	1.00
MAN 4730	Common Anode-red	.400	1.00
MAN 4740	Common Cathode-red	.400	1.00
MAN 4810	Common Anode-yellow	.400	1.00
MAN 6610	Common Anode-orange-D.D.	.560	1.25
MAN 6630	Common Anode-orange	.560	1.25
MAN 6640	Common Cathode-orange-D.D.	.560	1.25
MAN 6650	Common Cathode-orange ±1	.560	1.25
MAN 6660	Common Anode-orange	.560	1.25
MAN 6680	Common Cathode-orange	.560	1.25
MAN 6710	Common Anode-red-D.D.	.560	1.25
MAN 6730	Common Anode-red ±1	.560	1.25
MAN 6740	Common Cathode-red-D.D.	.560	1.25
MAN 6750	Common Cathode-red ±1	.560	1.25
MAN 6760	Common Anode-red	.560	1.25
MAN 6780	Common Cathode-red	.560	1.25
DL701	Common Anode-red ±1	.300	1.00
DL702	Common Cathode-red	.300	1.25
DL704	Common Cathode-red	.300	1.50
DL707	Common Anode-red	.300	1.50
DL741	Common Anode-red	.600	1.95
DL746	Common Anode-red ±1	.630	1.95
DL747	Common Anode-red	.600	2.25
DL749	Common Cathode-red ±1	.630	1.95
DL750	Common Cathode-red	.600	2.49
DL338	Common Cathode-red	.110	.69
FND70	Common Cathode (FND359)	.250	.75
FND503	Common Cathode (FND500)	.500	1.29
FND507	Common Anode (FND510)	.500	1.29
5082-7300	4 x 7 Sgl. Digit-RHDP	.600	19.95
5082-7302	4 x 7 Sgl. Digit-LHDP	.600	19.95
5082-7304	Overrange character (±1)	.600	15.00
5082-7340	4 x 7 Sgl. Digit-Hexadecimal	.600	22.50

CA3013	2.15	CA3082	2.00
CA3023	2.56	CA3083	1.60
CA3035	2.48	CA3086	.85
CA3039	1.35	CA3089	3.75
CA3046	1.30	CA3091	3.50
CA3053	1.50	CA3102	2.95
CA3059	3.25	CA3123	2.15
CA3060	3.25	CA3130	1.39
CA3080	.85	CA3140	1.25
CA3081	2.00	CA3401	.49
		CA3600	1.75

EXAR

XR-2206KB Kit $19.95 — **XR-2206KA Kit $14.95**

WAVEFORM GENERATORS		MISCELLANEOUS		TIMERS	
XR-205	$8.40	XR-2211CP	$6.70	XR-555CP	$.49
XR-2206CP	5.50	XR-4136	2.00	XR-320P	1.55
XR-2207CP	3.85	XR-1468	3.85	XR-556CP	1.85
STEREO DECODERS		XR-1488	5.80	XR-2556CP	3.20
XR-1310CP	$3.20	XR-1489	4.80	XR-2240CP	4.80
XR-1310EP	3.20	XR-2208	5.20	**PHASE LOCKED LOOPS**	
XR-1800P	3.20			XR-210	5.20
XR-2567	2.99			XR-215	6.60
				XR-567CP	1.95
				XR-567CT	1.70

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin	$.17	16	15	24 pin	$.38	37	36
14 pin	20	19	18	28 pin	45	44	43
16 pin	22	21	20	36 pin	60	59	58
18 pin	29	28	27	40 pin	63	62	61
22 pin	37	36	35				

SOLDERTAIL STANDARD (TIN)

14 pin	$.27	25	24	28 pin	$.99	90	81
16 pin	30	27	25	36 pin	1.39	1.26	1.15
18 pin	35	32	30	40 pin	1.59	1.45	1.30
24 pin	49	45	42				

SOLDERTAIL STANDARD (GOLD)

8 pin	$.30	27	24	24 pin	$.70	63	57
14 pin	35	32	29	28 pin	1.10	1.00	90
16 pin	38	35	32	36 pin	1.75	1.40	1.26
18 pin	52	47	43	40 pin	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

8 pin	$.40	.38	.35	22 pin	95	85	.75
10 pin	45	41	37	24 pin	$1.05	95	85
14 pin	39	38	37	28 pin	1.40	1.25	1.10
16 pin	43	42	41	36 pin	1.59	1.45	1.30
18 pin	75	68	62	40 pin	1.75	1.55	1.40

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

		10 OHM	12 OHM	15 OHM	18 OHM	22 OHM		
ASST. 1	5 ea.	27 OHM	33 OHM	39 OHM	47 OHM	56 OHM	1/4 WATT 5%	50 PCS.
		68 OHM	82 OHM	100 OHM	120 OHM	150 OHM		
ASST. 2	5 ea.	180 OHM	220 OHM	270 OHM	330 OHM	390 OHM	1/4 WATT 5%	50 PCS.
		470 OHM	560 OHM	680 OHM	820 OHM	1K		
ASST. 3	5 ea.	1.2K	1.5K	1.8K	2.2K	2.7K	1/4 WATT 5%	50 PCS.
		3.3K	3.9K	4.7K	5.6K	6.8K		
ASST. 4	5 ea.	8.2K	10K	12K	15K	18K	1/4 WATT 5%	50 PCS.
		22K	27K	33K	39K	47K		
ASST. 5	5 ea.	56K	68K	82K	100K	120K	1/4 WATT 5%	50 PCS.
		150K	180K	220K	270K	330K		
ASST. 6	5 ea.	390K	470K	560K	680K	820K	1/4 WATT 5%	50 PCS.
		1M	1.2M	1.5M	1.8M	2.2M		
ASST. 7	5 ea.	2.7M	3.3M	3.9M	4.7M	5.6M	1/4 WATT 5%	50 PCS.

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$5.00 Minimum Order — U.S. Funds Only
California Residents — Add 6% Sales Tax
Spec Sheets - 25¢ — Send 35¢ Stamp for 1978 Catalog
Dealer Information Available

NOW OPEN SATURDAYS

James ELECTRONICS

1978 CATALOG NOW AVAILABLE

1021-A HOWARD AVE., SAN CARLOS, CA. 94070
PHONE ORDERS WELCOME — (415) 592-8097
All Advertised Prices Good Thru January

J1

HOBBY-WRAP TOOL-BW-630

$34.95 (batteries not included)

- Battery Operated (Size C)
- Weighs ONLY 11 Ounces
- Wraps 30 AWG Wire onto Standard DIP Sockets (.025 inch)
- Complete with built-in bit and sleeve

WIRE-WRAP KIT — WK-2-W

WRAP • STRIP • UNWRAP

- Tool for 30 AWG Wire
- Roll of 50 Ft. White or Blue 30 AWG Wire
- 50 pcs. each 1", 2", 3" & 4" lengths — pre-stripped wire.

$11.95

WIRE WRAP TOOL WSU-30
WRAP • STRIP • UNWRAP - $5.95

WIRE WRAP WIRE — 30 AWG

25 ft. min. $1.25 50 ft. $1.95 100 ft. $2.95 1000 ft. $15.00
SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

WIRE DISPENSER — WD-30

- 50 ft. roll 30 AWG KYNAR wire wrap wire **$3.45 ea.**
- Cuts wire to desired length
- Strips 1" of insulation **Specify** — Blue-Yellow-White-Red

REPLACEMENT DISPENSER SPOOLS FOR WD 30
Specify blue, yellow, white or red **$1.98/spool**

DIP/IC INSERTION TOOL WITH PIN STRAIGHTENER
Inserts both 14 and 16 pin packages. Pin Straightener built into Handle.
Model INS-1416 **$3.49/ea.**

HP 5082-7400 SERIES — MULTI-DIGIT

			10 or more
• ⅛" ht. • Common Cathode Red	2 Digit	$ .79	.69
• 3-5 volts @ 5 mils/second	3 Digit	.89	.79
• 7 segment Monolithic	4 Digit	.99	.89
• Dip Package	5 Digit	1.19	.99

TV GAME CHIP SET — $9.95

Includes AY-3-8500-1 Chip and 2.010 mhz crystal
(2.010 crystal — $1.95 ea/AY-3-8500-1 Chip — $8.95 ea.)

ZENERS — DIODES — RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
1N746	3.3	400mm	4/1.00	1N4005	600 PIV	1 AMP	10/1.00
1N751A	5.1	400m	4/1.00	1N4006	800 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4007	1000 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N3600	50	200m	6/1.00
1N754	6.8	400m	4/1.00	1N4148	75	10m	15/1.00
1N959	8.2	400m	8/1.00	1N4154	35	10m	12/1.00
1N965B	15	400m	4/1.00	1N4305	75	25m	20/1.00
1N5232	5.6	500m	28	1N4734	5.6	1w	28
1N5234	6.2	500m	28	1N4735	6.2	1w	28
1N5235	6.8	500m	28	1N4736	6.8	1w	28
1N5236	7.5	500m	28	1N4738	8.2	1w	28
1N456	25	40m	6/1.00	1N4742	12	1w	28
1N458	150	7m	6/1.00	1N4744	15	1w	28
1N485A	180	10m	6/1.00	1N1183	50 PIV	35 AMP	1.60
1N4001	50 PIV	1 AMP	12/1.00	1N1184	100 PIV	35 AMP	1.70
1N4002	100 PIV	1 AMP	12/1.00	1N1185	150 PIV	35 AMP	1.50
1N4003	200 PIV	1 AMP	12/1.00	1N1186	200 PIV	35 AMP	1.80
1N4004	400 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR	$1.95
C38M	35A @ 200V	SCR	1.95
2N2328	1.6A @ 200V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC.	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC	1.95

TRANSISTORS

MPS A05	5/$1.00			PN4250	4/$1.00
MPS A06	5/$1.00			2N4400	4/$1.00
2N918	4/$1.00			2N4401	4/$1.00
2N2219A	3/$1.00	2N3638	5/$1.00	2N4402	4/$1.00
2N2221	4/$1.00	2N3702	5/$1.00	2N4403	4/$1.00
2N2222A	5/$1.00	2N3704	5/$1.00	2N4409	5/$1.00
2N2369	5/$1.00	2N3705	5/$1.00	2N5086	4/$1.00
2N2369A	4/$1.00	2N3706	5/$1.00	2N5087	4/$1.00
2N2484	4/$1.00	2N3707	5/$1.00	2N5088	4/$1.00
2N2906A	4/$1.00	2N3711	5/$1.00	2N5089	4/$1.00
2N2907A	5/$1.00	2N3724	$.65	2N5129	5/$1.00
2N2925	5/$1.00	2N3725	$1.00	2N5138	5/$1.00
2N3035	2/$1.00	2N3772	$2.25	2N5139	5/$1.00
2N3055	$.89	2N3903	5/$1.00	2N5209	5/$1.00
MJE2955	$1.25	2N3904	4/$1.00	2N5210	5/$1.00
MJE3055	$1.00	2N3905	4/$1.00	2N5432	$2.00
2N3392	5/$1.00	2N3906	4/$1.00	2N5449	3/$1.00
2N3398	5/$1.00	2N4013	3/$1.00	2N5951	5/$1.00
PN3567	3/$1.00	2N4014	3/$1.00	C106B1SCR	2/$1.00
PN3568	4/$1.00	2N4123	6/$1.00	40409	$1.75
PN3569	4/$1.00	PN4249	4/$1.00	40410	$1.75

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-49	50-100		1-9	10-49	50-100
10 pf	.05	.04	.03	.001µF	.05	.04	.035
22 pf	.05	.04	.03	.0047µF	.05	.04	.035
47 pf	.05	.04	.03	.01µF	.05	.04	.035
100 pf	.05	.04	.03	.022µF	.06	.05	.04
220 pf	.05	.04	.03	.047µF	.06	.05	.04
470 pf	.05	.04	.035	.1µF	.12	.09	.075
100 VOLT MYLAR FILM CAPACITORS							
.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22
+20% DIPPED TANTALUMS (SOLID) CAPACITORS							
.1/35V	.28	.23	.17	1.5/35V	.30	.26	.21
.15/35V	.28	.23	.17	2.2/25V	.31	.27	.22
.22/35V	.28	.23	.17	3.3/25V	.31	.27	.22
.33/35V	.28	.23	.17	4.7/25V	.32	.28	.23
.47/35V	.28	.23	.17	6.8/25V	.36	.31	.25
.68/35V	.28	.23	.17	10/25V	.40	.35	.29
1.0/35V	.28	.23	.17	15/25V	.63	.50	.40
MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS							
Axial Lead				**Radial Lead**			
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.18	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.15	.13	.10	1.0/16V	.15	.13	.10
4.7/25V	.16	.14	.12	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.18	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.18	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.18
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

California Industrial
Post Office Box 3097 K • Torrance, California 90503
CALCULATOR
WRIST WATCH
3.1415927
Designed for the on the go executive, that individual who has to make those on the spot decisions.
Handsome gold tone stainless steel watch features space age micro-circuitry. The MOS inte grated circuit contains the equivalent of more than 10,000 transistors.
This LED wrist watch displays date, time, elapse seconds and also functions as an eight digit calculator with memory. Information stored in memory can be recalled at any later date, even weeks or months. Use this memory feature to store phone numbers, parking stall location or flight departure time.
Manufactured by one of California's leading aerospace contractors. Because of the discount price we have agreed not to publish the manufacturer's name.
Includes batteries, jewelry case and 18-month factory warranty.
$149.50
ELECTRONIC ENTERTAINMENT CENTER
Tennis·Handball
Hockey·Smash
Action-packed color entertainment for the whole family. Adjustable skill level controls allow players of all ages to compete in tennis, hockey and handball. This four game entertainment center turns your television into a video playground.
On screen scoring, live action sound and true component color makes this video center an excellent buy at only $24.88.
Complete with antenna box and AC adapter.
Color $24.88
TELETYPE® MODEL 43
New from Teletype, the Model 43 is capable of printing 132 ASCII characters per line. Send and receive data at 10 or 30 char. per second. Keyboard generates all 128 ASCII code combinations. RS-232 interface, same as the popular Model 33. Data sheet sent upon request. Manufacturer suggested price $1377.00.
IMMEDIATE DELIVERY $1199
We also have for sale a limited quantity of used Model 35's. Priced at only $449.50
TELETYPE
BARREL STOCK tm
TEST 'EM YOURSELF 'N SAVE
15 .... NE555 Timers, 100% functional
15 .... TRW Power Darlingtons, TO:3
15 .... 2N3055 & 2N3772 NPN power trans.
50 .... General Purpose TO:92 Transistors
100 .... Silicon Signal Diodes
$195
Abe's barrel
SPECIAL
GENERAL INSTRUMENT ASCII Keyboard Encoder AY 5-3600 Prime but house marked only $4.95
DIGITAL
ALARM CLOCK
Completely Assembled $19.95
3 for $50
10:26
UNIVERSAL
Walnut-grained decorator clock features large .7" LED display which is driven by the new National MM5385 alarm clock chip. Preset 24-hour alarm function allows you to awaken at the same time each morning without resetting. Upon reaching the wake-up time, the clock's loudspeaker emits a gentle tone. Touch the snooze button and doze off for an additional 9 minutes of sleep. Clock also functions as a ten-minute elapse timer. "Alarm Set" Indicator, AM-PM display.
CONNECTORS
HOOD
RS-232
DB25P male plug & hood
$3.95
MALE
DB25S female
$3.95
FEMALE
Edge Connectors
100 PIN
IMSAI/ALTAIR
S-100 • GOLD PLATED • .125" CENTERS
Altair .140 row, soldertail..... $5.98 3/$16.50
Imsai .250 row, soldertail..... $4.98 3/$13.00
3 Level Wire Wrap .250 row... $4.98 3/$13.00
SPECIALS
W/W same as above without ears$3.50 3/$10
72 (dual 36) W/W .156" centers... $2.50 3/$6
$24.88
SPERRY UNIVAC
KEYBOARD
The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88. The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry.
Original cost was $385. Used but guaranteed in excellent condition. Complete with documentation.
Quiet Buss™
S-100 MOTHER
The Quiet Buss from California Industrial is quality engineered. No short cuts have been taken to produce this mother board. Active termination circuitry prevents noise and crosstalk. Manufactured from extra heavy FR-4 epoxy glass. Features 2 ounce double thickness copper traces. Purchase this board along with the power supply kit below and you have the start of a super Micro-System.
18 SLOT 29.95
FREE
MANUAL GRAPHITE DISPLAY GENERATOR
Modern technology has pioneered the development of this unique character printer. Our Manual Graphite Display Generator has the capability of producing the full upper and lower case ASCII set. Self-contained cursor assembly allows the operator to eliminate erroneously entered information.
Each unit is manufactured to strict tolerances as prescribed by standards set forth by California Industrial. One free with every order.
$4.98
10 for $45.
Scotch® BRAND
DISKETTES
Please specify IBM 3740 series or 32 sector.
Certified Digital CASSETTES
Won't drop a BIT!
$4.98
JOYSTICK $4.50
3 for $10.00
This joystick feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Perfect for television games, quad stereo and radio controlled aircraft.
intel 2708
450ns
$14.95
8K UV Erasable MEMORY
SUPERECTIFIER
$1.95
MDA 980-1
BRIDGE RECTIFIER
MOTOROLA 12 Amp. 50v.
990-1 27 Amp. $2.95
R.F. MODULATOR
from Atari
$13.95
Converts signal from APPLE II and other vidio sources into TV frequency. Complete with directions and metal case. Assembled
21L02
LOW POWER
450 NS
$1.19
SUPER BUY!
Lowest Price Anywhere
Our low power static RAMs are factory prime. Purchased, on contract, directly from one of California's leading semiconductor manufacturers.
Access time guaranteed faster than 450nS. Minimum purchase 32 pieces.
Sorry, credit cards can not be accepted on 21L02's.
7400 .13
7401 .19
7402 .19
7403 .19
7403 .19
7404 .19
7405 .19
7406 .19
7407 .25
7408 .25
7409 .25
7410 .19
7411 .25
7412 .35
7413 .49
7414 .79
7416 .39
7417 .39
7420 .19
7422 .49
7423 .39
7425 .39
7426 .39
7427 .39
7428 .49
7429 .39
7430 .25
7433 .39
7437 .39
7438 .39
7439 .39
7440 .39
7442 .59
7443 .79
7445 .89
7446 .99
7447 .99
7448 .99
7450 .25
7451 .25
7453 .25
7454 .25
7460 .25
7470 .25
7472 .39
7473 .39
7474 .35
7475 .49
7476 .39
7479 3.99
7480 .79
7482 .99
7483 .99
7485 .99
7486 .49
7488 3.40
7489 2.79
7490 .49
7491 .99
7492 .49
7493 .49
7494 .79
7495 .79
7496 .79
7497 3.99
74100 1.19
74107 .39
74109 .49
74110 1.79
74116 1.99
74120 1.79
74121 .39
74122 .39
74123 .69
74125 .59
74126 .59
74128 .49
74132 .99
74136 .89
74141 .99
74145 .99
74147 2.49
74148 1.99
74150 1.19
74151 .99
74153 .89
74154 .99
74155 .99
74156 1.29
74157 .99
74159 2.99
74160 1.19
74161 .99
74162 1.49
74163 .99
74166 1.19
74167 4.99
74170 2.49
74173 1.49
74174 1.19
74175 .99
74176 .99
74177 .99
74192 .99
74193 .99
CMOS
4001 .25
4002 .25
4006 1.99
4007 .25
4008 1.49
4009 .69
4010 .69
4011 .25
4012 .25
4013 .49
4014 1.49
4015 1.39
4016 .69
4017 1.29
4018 1.69
4019 1.79
4020 1.39
4021 1.49
4022 1.25
4023 .25
4024 1.19
4025 .25
4027 .69
4028 1.25
4029 1.99
4030 .69
4032 .49
4033 1.99
4035 1.69
4040 1.99
4041 1.49
4042 1.49
4043 1.49
4044 1.49
4046 2.49
4047 2.49
4049 .79
4050 .79
4051 1.99
CPU's
8080A 14.95
6800 24.95
Z80 39.95
MEMORY
1702A 4.95
82s23 2.95
82s123 2.95
2102 1.79
2102-1 1.89
21L02 1.19
CLOCK's
5314 2.95
5316 4.95
5375 3.95
LM741
$.09
300H .79
301H .39
301CN .39
302H 1.29
304H 1.29
305H .99
307H .49
307CN .39
308H .99
308CN .99
309H 1.09
309K .99
310H 1.19
310CN 1.19
311H .99
311N .99
312H 1.99
318H 1.79
318CN 1.49
319CN 1.29
320K-5 1.39
320K-12 1.39
320K-15 1.39
320T-5 1.75
320T-8 1.75
320T-12 1.75
320T-18 1.75
320 -24 1.75
324N 1.79
339N 1.69
340K-5 1.95
340K-8 1.95
340K-12 1.95
340K-15 1.95
340K-18 1.95
340K-24 1.95
340T-5 1.79
340T-12 1.79
340T-15 1.79
340T-24 1.79
350N .99
351CN .65
370H 1.29
370N 1.29
373N 3.19
377N 3.99
380N 1.39
381N 1.79
382N 1.79
NE555v .49
NE556 1.29
NE565H 1.49
NE565N 1.79
NE566N 1.25
703CN .45
709H .39
709N .39
710N .79
711H .39
711N .39
723H .55
723N .55
725H 3.49
733H 1.49
733N .99
739N 1.19
741N .09
747N .79
748H .39
748N .39
1414N 1.75
1458 .69
1496N .95
2111N 1.95
2901N 2.95
3065N .69
3900N .49
3909N 1.19
factory prime
2114/4045 1K by 4 STATIC MEMORY
450nS. $11.95 650nS. $9.95
MICROCOMPUTER POWER SUPPLY KIT
Ideal supply for S-100 systems. Contains all the necessary components to produce 5 volts @ 10 A., ± 12 @ 2.5 A. (voltage actually higher). New surplus from Memorex Corporation.
Kit includes 117 ac. transformer, diodes, six computer grade electrolytics, chassis, circuit breaker and grounded power cord. Transformer has 5 output windings.
Include $9.00 shipping East of the Mississippi; $4.00 for Calif.; all other States $7.00.
$24.88
Plus Shipping
SPEAKERS
$.69
6 for $3.00
2¼" Miniature
Perfect for TV games, alarm systems, speech synthesizer or intercom. 8 ohms
5" HiFidelity $2.98
Thumbwheel switch
Ten position
BCD
$1.39 ea.
10 50
$1.19 .89
EECO
MINIATURE SWITCHES
CK
your choice
10 50 100 1k
$.98 $.88 .81 .73 .66
SPDT Miniature Toggles
7101 C&K ON-NONE-ON
7107 jbt ON-OFF(mnt.ON)
7108 CK ON-(moment.ON)
Rocker JBT DPDT
Rotary 3P-4-Pos.
Rotary 3P-6-Pos.
Push B (N.O.) $.39ea. 3/$1
TRIMMER POTENTIOMETERS
2K 5K 10K 50K
5 for $.98
20 50 100
16¢ 14¢ 12¢
CAPACITORS
ELECTROLYTICS
ea. 10 50
4500/50v. $1.49 1.35 1.19
1000/15v. axial $.55 .49 .45
.1 disc $.12 .09 .07
.01 disc .06 .05 .04
4500
DIP Switch
$1.75 ea.
10 25 100
$1.63 1.52 1.29
specify 7 or 8 pos.
OPEN
SPST
32 Conductor
$.59 Ft.
RIBBON WIRE
SPECTRA-STRIP
Power Adapter
6 vdc, 140mA $1.39
7 dc, 1.4 A. 5.50
9 vdc, 15mA. 1.19
9 vdc, 175mA. 1.95
12 vdc, 600mA 2.95
RELAYS
SPDT MINIATURE
$1.39 ea.
10 25 100
$1.15 1.04 .89
Coil 12 Volt dc.
7 Amp Contacts
P.C. Board Mount
Transistors
ea. 10 50 100
2N2222A .20 .18 .16 .15
2N3055 .69 .65 .59 .55
MJ3055 .79 .75 .69 .65
2N3772 1.59 1.49 1.39 1.29
2N3904 .15 .11 .09 .07
2N3906 .15 .11 .09 .07
Diodes
10 25 100
1N4002 100v. .08 .06 .05
1N4005 600v. .10 .08 .07
1N4148 signal .07 .05 .04
jumbo red LED's
ea. 10 25 100
$.15 .13 .11 .09
$.98
Output:
12v.ct. 175mA.
TRANSFORMER
SOLAR CELLS
$1.19
10 100
$.98 $.85
125mA. at .43V. Ideal for powering radio receivers or charging batteries.
9 foot
$1.49
Heavy duty grounded power cord and mating chassis connectors.
$.98
Nicad
500mA "AA"
FAST CHARGE
DISCOUNT
Wire Wrap Center
IC SOCKETS
wire wrap / low profile
pin ea. 25 50 ea. 25 50
8 17¢ 16 15
14 37¢ 36 35 18 17 16
16 38 37 36 19 18 17
24 99 93 85 36 35 34
40 1.69 1.55 1.39 63 60 58
50ft. $.98
KYNAR WIRE WRAP
500 1,000 11,000
$9. $15. $105.
Battery wire wrapping tool
$34.95
$29.95
OK HOBBY WRAP-30 wire wrap & strip tool
$5.45
(213) 772-0800
VISA
master charge
All merchandise sold by California Industrial is premium grade.
Orders are shipped the same day received.
PLEASE INCLUDE $1.00 SHIPPING ON ORDERS UNDER $ 15.00
California residents add 6% sales tax • Money back guarantee.
Sorry, no COD's • Foreign orders add 10%
Credit cards accepted, $20 minimum
Reader inquiry card number C·50

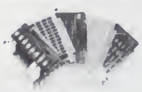
ADVANCED COMPUTER PRODUCTS

THE FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST, AT FAIR PRICES NOW LOWERS PRICES EVEN FURTHER!

1. Proven Quality Factory tested products only, no re-tests or fallouts. Guaranteed money back. We stand behind our products.

2. Same Day Shipment All orders will be shipped same day as received.

MOTOROLA 6800 COMPATIBLE MODULES



MEK6800 D2 KIT $235.00

9601 16 Slot Mother Bd.	$175.00
9602 16 Slot Card Cage	72.00
9610 Proto Board	36.00
9615 4K EPROM Module (1702A)	350.00
9615K 4K EPROM Kit Module	275.00
9620 16 Port Parallel I/O	375.00
9626 8K Static RAM Module	350.00
9626K 8K Static RAM Kit	275.00
9650 8 Port Duplex Asyn Serial I/O	395.00
Connectors $6.50	16/72.00

All assembled & tested not Kits

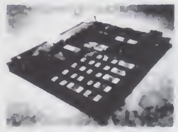
PLUS MOTOROLA TV MONITORS-PRIME

Model M3003-L01 12" display	$219.95
Model M2000-155 9" display	$199.95

Add $10.00 for shipping

AMI EVK 99

only $133.00



EVK 99 Kit	$133.00
EVK 99 EX expanded	449.95
Universal Kluge Board	95.00
16K Byte RAM Board	75.00
6 Slot Motherboard	35.00
Extender Board	45.00
Video Board	95.00
Connectors . . $6.50	12/59.95
Solid Frame Chassis	120.00
Frame Chassis	69.95
Micro Assembler ROM	30.00
Proto ROM	30.00
Tiny Basic Papertape	20.00
Tiny Basic EPROM	125.00

RAM'S (cont.)

4096 16 pin	4.95
2107B-4	4.25
MM5261	3.95
MM5262	1.95
MM5280	4.95
8599	1.88
P3101	2.95
5101C-E	9.95
3107	3.95
1101	1.95
1103	1.95
4008 L	2.95

FLOPPY DISK

MPD 372	$55.00
WD1771 Floppy	59.95

Application notes included

CHARACTER GENERATORS

R032513 Upper	$10.95
R032513 Lower	9.95
HD0165	9.95
MC6571	14.95
MC6574	14.95
MC6575	14.95

KEYBOARD ENCODERS

AY52376	$14.95
AY53600	14.95

UARTS/USRTS

AY51013A (5V)	$ 5.50
TR1602B (5V)	5.50
AY51014 (5-14V)	8.95
AY51015 (5V)	9.95
IM6402 (5V)	12.95
IM6503 (5V)	12.95
S2350	10.95
WD1671B Astros	29.95

BAUD RATE GENERATORS

MC14411	$11.95
WD1941 Dual	9.95
34702	13.95

TV CHIPS

MM5320 TV Synch.	$ 7.95
MM5369 Prescaler	3.95
MM57100 Game Chip	16.95
MM57104 Clock	3.75
LM1889 Modulator	3.95
CW300 Saw Function	1.50
AY38500-1 TV Game	9.95
AY38600-1 Color TV Game	24.95
AY38615-1 Color Converter	8.95
AY38700-1 Tank Chip	29.95

ORGAN CHIPS

MM5554 / MM5555 / MM5556	NSC Set $19.95

CLOCK CHIPS

MM5313	$4.50
MK 50250	4.50
MM5314	4.95
MM5316	4.95
MM5375	4.95

A TO D CONVERTORS

8700CJ	$13.95
8701CN	21.95
8750CJ	13.95
1408L8	5.95

DISPLAYS/LEDS

FND 357 (.375)	$1.25
FND 367 (.375)	1.35
FND 500 (.500)	1.25
FND 507 (.500)	1.00
FLV 110 Redled	5/1.00
FND 800 (.800)	2.50
FND 807 (.800)	2.50
HP 7730 7 Segment	1.50
HP 7340 Hex Display	15.95

UV EPROM ERASER NEW LOW-COST MODEL ERASES 4 PROMS AT ONE TIME $59.95

SOCKETS

14 Pin ww	.37
16 Pin ww	.38
18 Pin ww	.60
22 Pin ww	.89
24 Pin ww	.85
40 Pin ww	1.25
8 Pin S/T	.17
14 Pin S/T	.20
16 Pin S/T	.22
18 Pin S/T	.31
22 Pin S/T	.35
24 Pin S/T	.41
28 Pin S/T	.49
40 Pin S/T	.63

NEW CTS DIPSWITCHES

CTS206-4	$1.75
CTS206-5	$1.75
CTS206-6	$1.75
CTS207-7	$1.75
CTS208-8	$1.95
CTS209-9	$1.95
CTS209-10	$1.95

NEW CATALOG NOW AVAILABLE

CONNECTORS

S100 BUS W/W IMSAI	$5.00
S100 BUS S/T IMSAI	5.00
S100 BUS W/W ALTAIR	5.50
S100 BUS S/T ALTAIR	5.25
RS232 25 Pin Male DB25P	3.50
PS232 25 Pin Female DB25S	3.95
Plastic Hood	.75
IMSAI Card Guides	4/1.00

RS232 CONNECTOR SET WITH HOOD 6.50

CRYSTALS

1.0MH$_3$	$ 5.85	10.0MH$_3$	4.95
2.0MH$_3$	5.85	13.0MH$_3$	4.95
2.097152MH$_3$	5.85	14.31818	4.95
2.4576MH$_3$	5.85	18.0MH$_3$	4.95
3.579545MH$_3$	3.95	18.432MH$_3$	5.95
4.0MH$_3$	4.95	20.0MH$_3$	4.95
4.194304MH$_3$	5.95	22.1184MH$_3$	5.95
4.91520MH$_3$	5.95	27.000MH$_3$	5.95
5.0MH$_3$	4.95	36.000MH$_3$	5.95
5.7143MH$_3$	5.95	48.000MH$_3$	5.95
6.0MH$_3$	4.95	100KC	12.95

MICROPROCESSORS

Z-80	$24.95
Z-80A	34.95
F-8 (3850)	19.95
2650	24.95
CD1802	19.95
8080A	12.95
8085	29.95
8008-1	14.95
2901	21.95
2901A	29.95
TMS 9900JL	89.95
CP1600	39.95
6502	19.95
IM6100	29.95
6800P	24.95
6802P	32.95

SUPPORT DEVICES

3881 (Z80 PIO)	$12.95
3882 (Z80 CTC)	12.95
3851 (F8 Program	14.95
3853 (F8 Memory I/O)	14.95
8212 8 bit I/O	3.50
8214 Priority Interrupt	9.95
8216 Bus driver	3.75
8224 Clock Gen.	5.95
8224-4 4MHz CLK Gen.	12.95
8T26 Busdriver	2.25

SUPPORT DEVICES (cont.)

8226 Busdriver	3.95
8228 Sys Controller	7.95
8238 Sys Controller	9.95
8251/9551 Prog. Comm. I/O	11.95
8253 Interval Timer	21.95
8255/9555 Prog. Periph. I/O	11.95
8257 DMA Control	32.95
8259 Prog. Interrupt	32.95
6810-1 128 x 8 RAM	4.95
6820 PIA	8.95
6834-1 EPROM 4K	14.95
6834 EPROM 4K	16.95
6850 ACIA	9.95
6852 Serial Adapter	11.95
6860 Modem	14.95
6862 RPS Modulator	17.95
6871B 1.0 MHz Clock OSC	26.95
6880 MPU Busdriver	2.95
68 MinII Minibug 2	49.95
68 MinIII Minibug 3	74.95
6831/10224 Micro Assembler Rom	30.00
6831/11003 Proto Rom	30.00
1821SCD 1K RAM	15.95
1822SCD 256 x 4 RAM	16.95
1824CD 32 x 8 RAM	9.95
1852CD 8 bit I/O	10.95
1856CD I/O	8.95
1857CD I/O	8.95

PROM'S

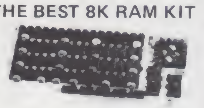
2708	$19.95
1702A	7.95
1702-6	4.49
5204	16.95
6834	16.95
6834-1	14.95
82S126	4.50
8223	2.95
5203	6.95

Programming Available

RAM'S

21L02 (450ns)	$1.75
21L02 (250ns)	2.25
2102(450ns)	1.69
2102(650ns)	1.59
2102(1 sec)	1.40
2111	3.95
21L11	4.25
2112	2.95
2101	2.95
21L01	3.95
74C89	4.95
74S189	2.95
7489	1.88
4116 16K	34.95
TMS4060 22 pin	4.25
WD4060 22 pin	3.25

LOGOS I THE BEST 8K RAM KIT



Assembled & Tested $199.95
Kit Price Only $149.95

Features:

Lowpower 21L02, 450ns, Dip Switch Selectable addressing down to 256 Byte blocks, No wait states, fully buffered, battery back-up.

★ Plus the first 32K Static Memory Board Uses 2114 4K RAMS ★

16K	$549.00
24K	729.00
32K	945.00

DATA BOOKS

NSC Digital	$ 3.95
NSC Linear	4.95
NSC Linear A/N Vol. I	2.95
NSC Linear A/N Vol. II	2.95
NSC CMOS	2.95
NSC Audio	2.95
NSC Voltage Reg	2.95
NSC Memory	3.95
Intel Data Book	4.95
NSC Special Functions	3.25
NSC Mos/LSI	4.25
NSC Transistors	2.95
Iasis Microprocessor Handbook	7.95
Iasis Complete up Learning Course	99.50
Intel MCS80 Manual	4.95
Intel MCS40 Manual	4.95
AMI MOS Catalog	2.95
Raytheon Linear	2.95
Raytheon Quads/Duals	1.95
New GI mos	4.95
Osborne Introto Microputer Vol. I	7.50
Osborne Introto Microputers Vol. II	12.50
Osborne 8080 Programming for Logic Design	7.50
Osborne 6800 Programming	7.50

SPECIALS

MCT2	$ .89
2N3638A	5/1.00
2N3640	5/1.00
2N2369A	5/1.00
2N3692	5/1.00
MSPS4318	5/1.00
MPS5401	5/1.00
MPS6516	5/1.00
MPS6544	5/1.00
MPS3568	5/1.00
MPS6522	5/1.00
MPS404A	5/1.00
2N5462	5/1.00
MPS3642	5/1.00
MPS6571	5/1.00
MPS3693	5/1.00
MPS6530	5/1.00
MPS3646	5/1.00
7525J	.50
7524J	.60
74S258	.60
7489	1.49
74145	.60
74H55	.19
715DC	1.49
74199	1.00

Plus huge backup inventory with special pricing for VOLUME requirements. Call for a quotation

BRIDGE SPECIAL

MDA 952-2	$1.49
MDA 962-3	1.49
KBPC2504	1.95

FREE 1977 CATALOG 56 pages
Send .25 for postage

COMPUTER KITS & SYSTEMS

MEK6800D2 Kit	$235.00
KIM1 6502	245.00
EVK99 6800 Kit	133.00
Intercept Jr. 6100 Kit	281.00
Iasis 8080 Computer Book	499.00
NSC SC/MP Kit	99.00
NSC Keyboard Kit	95.00
Low Cost S-100-BUS with 8 slots & Power Supply	159.95
8K Ram Kit (Logos I)	149.95
Byteuser 8K Eprom Kit	64.95
ZPU Kit (Z80)	295.00
TDL ZPU Kit (Z80)	269.00
Tarbell cassette I/O	115.00
S100 8 slot Motherboard with connectors (expandable)	69.95
S100 Extender Board	15.95
S100 Proto Board	27.95
Vector 8800 Proto Board	19.95
IMSAI 8080 w/22 slot	699.00
Cromemco Z-2	595.00
TDL XITAN alpha 1	769.00
Computalker	385.00
Hevristics Speechlab	299.00
SOROC IQ120 Terminal	995.00
8K Ram Board (Logos)	28.95
8K Eprom Board	24.95
S100 32K Ram Kit	945.00
Bytesaver Kit	145.00

All Shipments FCM or UPS. Orders under $100.00 add 5% handling and postage. Orders over $100.00 add 2.5% handling & postage. Mastercharge/Bankamericard/COD accepted w/25% deposit. California Residents add 6% tax. Foreign Orders add 8% handling. All parts prime factory tested guaranteed. Same day shipment. Add 25 cents for Data.

LOWEST PRICES YET

2708 Eprom 450 HS	$19.95
2716 16K Eprom	36.95
MK4116 (upD416) 16K Ram	34.95
Intel 3404 6 bit latch	2.95
2114 4K Static	12.95
upD410 4K Static	11.95
4200A 4K Static	12.95
1488 or 1489	1.25
8T97	1.25
8130	2.45
8131	2.45
MH0026CN	2.50
8833	1.85
8835	1.85
74367/74368	6/5.00
WD 4060 4K Ram	3.25
1702A-6	10/32.95
75452	10/2.50
8T26	2.25
H RAM 64 bit Ram	.75
78L05 Volt. Reg.	only .59
LM318H	1.00

BYTEUSER 8K EPROM WITH POWER ON JUMP/RESET JUMP USES 2708

Byteuser Kit	only $ 64.95
PCB only	24.95
8K Eprom ((8)(2708))	125.00

SPECIALTY CHIPS

MPD371 Mag. Tape Control	$49.95
AYS 3550 4¾ DVM	24.95
MK5007/MK5009 Counter	6.95
AYS 3507 DVM	12.95
ICM7208IPI Ctr./Disp/Driver	16.95
ICM7045IPI Stop Watch	18.95
G115M, Intersil Analog Switch	1.50

Retail pricing may vary from Mail Order Pricing.

ADVANCED COMPUTER PRODUCTS

P. O. BOX 17329 K
Irvine, California 92713
New Phone (714) 558-8813

★ NOW Our First Store Open
1310 B East Edinger
Santa Ana, CA 92705 ★

Triplett Model 3300

The "Hand-Size" Triplett Model 3300 is a battery operated DVOM weighing only 10 ounces. Its combination of small size, light weight and battery operation makes it ideal for those "away from workbench" trouble-shooting and measurement requirements. This 5 function, 22 range DVOM includes Hi/Low power ohms, auto-zeroing, auto-polarity and out-of-range display blinking. With the AC adapter charger, the tester has the convenience of battery or AC operation. Whatever your measurement needs, see the new Model 3300 DVOM at TRI-TEK!!!

*3½ Digit High intensity .3" LED display.
*5 functions, 22 ranges, single range selector switch.
*Single range selector switch is color coded.
*Typical DC accuracy is .5% Rdg.
*6 resistance ranges to 20M, 3 low power ohms.
*Overload protection up to 600V on all ranges.
*Flashing overrange indication on display.
*Auto-polarity.
*One year warranty
*Offset zera adjustment capability.
*Convenient "Snap - in battery pac.
*Includes Ni Cad batteries and AC adapter charger.
*Single polarized test lead plug.
*High impact thermoplastic case.

MODEL 3300.............................$175
Cat #10-2860 optional carrying case.........$16

Charge card telephone orders ($20 min.) will be accepted 9-5:30 P.M. except weekends. Telephone 995-9352. No collect calls please.

TRIPLETT

GENERAL MULTI-PURPOSE V-O-M
*Drop-resistant, hand-size V-O-M with high-impact thermoplastic case.
*20,000 Ohms per volt DC and 5,000 ohms per volt AC; diode overload protection with fused Rx1 ohms range.
*Single range switch; direct reading AC Amp range to facilitate clamp-on AC ammeter usage.

RANGES

DC Volts: 0-3-12-60-300, 1,200 (20,000 ohms/volt.)
AC Volts: 0-3-12-60-300-1,200 (5,000 ohms/volt).
Ohms: 0-20K-200K-2M-20M(200 ohm center scale on low range).
DC Microamperes: 0-600 at 250mV
DC Milliamperes: 0-6-60-600 at 250mV.
Accuracy: ±3% DC; ±4% AC; (full scale).
Scale length; 2-1/8"
Meter: Self-shielded; diode overload protected; spring backed jewels.
Case: Molded, black, high impact thermoplastic with slide latch cover for access to batteries and fuse. 2-3/4" w x 1-5/16" d x 4-1/4" h.
Batteries: NEDA 15V 220 (1), 1½V 910F (1):

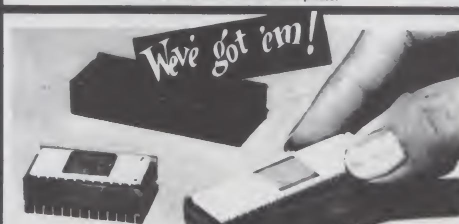
Complete with 42" leads, alligator clips, batteries and instruction manual.
MODEL 310.............................. $55
Cat # 10-1258 Optional carrying case........ $6.40

INTEGRATED CIRCUITS

555 Timer 8 pin mini-DIP	.49
741 Compensated OP-Amp 8 pin DIP	.37
LM 1889N RF Video Modulator	7.45
CA3130 Bipolar/Mos-FET Op Amp	1.19
CA3140 MOS-FET Op Amp, Bi-polar out	.99
LM3909 Lo Voltage Led Pulser	1.50
LM3911 Temp Control CHIP	1.50
Signetics 2504TA 1024 bit dynamic SR memory	1.75
MCM 6571P Character Generator	9.95
MCM6571AP Character Generator	9.95
MC14409P Telephone Rotary Pulser	10.98
MC14419P Touch Pad Converter for 14409	4.25
MC14411P Baud Rate Generator	11.98
MC14412VP CMOS Modem Chip	16.95
MM57109N Number Cruncher Micro	18.95
74C915 7 Segment to BCD Converter	2.99
74C922 16 Key Keyboard Encoder	6.35
74C923 20 key Keyboard Encoder	6.45
74C925 4 Decade Counter w/latches	12.00
74C926 4 Decade Counter w/carry	12.00
74C935-1 3½ Digit DVM CMOS Chip	16.98
TL170 TO-92 Hall effect switch w/spec sheets	1.25
LH0070-1H Precision (.3%) 10V Reference Amp	5.35
LHOO70-2H Extra Precise (.05%) 10V Reference Amp	10.55
LM399H Temp Stabilized Zener	5.95
AF100-1CJ Active Filter, State Variable	7.50
LM2907N Tachometer F/V Converter	2.65
LM1812N Ultra-sonic Transceiver	9.15
LM1815 Adaptive Sense Amp for Tachometer	5.73

tri-tek, inc.
7808 North 27th Avenue
Phoenix, Arizona 85021
(602) 995-9352

- We pay surface shipping only in USA, Canada and Mexico.
- For premium shipping (first class, special handling, etc.) add extra. Excell will be refunded.
- Foreign orders (except Caneda and Mexico) estimate and add shipping. Excess will be refunded.
- Please give street address for UPS shipping when possible.
- C.O.D. NO parcel post C.O.D.
  UPS C.O.D. Add 85¢ to order.
- Any correspondence not connected with your order, please use separate sheet and include SASE for reply.
- Orders less than $10 ($15 foreign) please add $1 handling.
- Prices are subject to change without notice.
- Any refunds will be by check, not credit vouchers.
- If we should be temporarily out of stock on an item, it will be placed on back order. If we cannot ship in 30 days, you will be notified of the expected shipping date and furnished with a postage paid card with which to cancel your order if desired.
- Terms. Check, money order, credit card. Net 30 days to rated firms, schools and government agencies.



ZIF-14	$3.20	ZIF-22	4.50
ZIF-16	3.49	ZIF-24	4.85
ZIF-18	3.95	ZIF-40	7.19

(Last two digits of part number show number of pins per socket)

Unique Features of the Welcon Zero Insertion Force Dual-In-Line Sockets . . .

- **Capability to accept devices with up to .070" lead spread variation.**
- **Accepts lead lengths as short as .075".**
- **Sturdy terminals that fit .025" diameter printed circuit board holes.**
- **"Zero" insertion, positive wiping contacts.**
- **"ZIF" eliminates costly damage to fragile LSI devices**

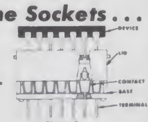


Figure 1
Device before insertion.

Why struggle to insert DIP's into sockets when you can do it the easy way . . . the Welcon way . . . with a new design concept that provides "zero insertion force" and positive wiping.

The "604" series of sockets, with all of its positive features and higher reliability, is small in size. High density packaging can be achieved for burn-in and production.

OPERATION: Lid is moved to up position stop (Fig. 2). This cams contacts into "open" position. DIP device can then be dropped into open contacts. In Fig. 3 DIP is pushed downward and contacts begin to close. When tip of device lead is past contact point (Fig. 4), contacts close and wipe on lead for remaining distance.

Figure 2

Figure 3

Figure 4

VOLTAGE REGULATORS

7805-06-08-12-15-24 TO220	95¢ 5/$4.50
7905-06-08-12-15-24 TO 220	95¢ 5/$4.50
78LO5A-12-15 4% 100 mA TO-92 Plastic	50¢
78HO5KC 5V 5A TO-3	9.15
78H12KC 12V 5A TO-3	9.15
78H15KC 15V 5A TO-3	9.15
Lm317K 1.5A *Adjustable* TO-3	4.99
Lm317T 1.5A *Adjustable* TO-220	3.99
Lm317MP .5A *Adjustable* TO-202	13.95
TL430C Adjustable Zener-*Think About It*	1.50
TL497C Switching Reg. & Inductor	9.50
RCA CA 3085 100 mA *Adjustable*	.60

DIODES AND BRIDGES

IN4003 200 V 1 amp	12/$1.00
IN4004 400 V 1 amp	10/$1.00
IN4148 Hi Speed Signal	15/$1.00 100/$5.00
D-600 115 V, 100 mA Hi Speed Signal	20/$1.00
D2131 200 V, 25A Stud	85¢
D2135 400 V, 25A Stud	1.00
D2138 600 V, 25A Stud	1.55
D3289R 200 V, 160A Stud Anode	5.85
D3909-4 50 V, 45A Fast Recovery	2.00
IN4732A-47A 1W 5% Zeners	4/$1.00
13 Assorted *Brand New* Zener Diodes	1.00
50V 3 amp Epoxy Bridge	79¢
200V 30 amp Bridge	2.00
600V 4 amp Epoxy Bridge	1.49
600V 3 amp Stud Bridge	.89
SI-2 200V, 1.5A Gold Leads	15/$1.00
D1A-0030 30V DIAC	10/$1.00

I.C. SOCKETS

Lo Profile Tin Solder Tail Dip Sockets

8 pin	10/$1.50	100/$14.00	1000/$120.00
14 pin	10/$1.70	100/$16.00	1000/$140.00
16 pin	10/$1.90	100/$18.00	1000/$160.00

6.3 VCT, 1.2A Transformer F41X	2.49
12V, 1A Transformer with 6' Power Cord	2.88

MISCELLANEOUS

RG-174 Miniature 50 Ω coax	50'/4.25
WSU-30 Wire Wrap/unwrap tool	5.95
WSU-30M Modified Wrap/unwrap tool	6.95
BW-630 Battery Operated Wrap Tool	34.95
—Free Wire with any Wrap Tool—	
Miniature Square .05/100V Monolithic Cap	10/2.00

T1

Bearcat® 210

$289.

The Bearcat® 210 is a sophisticated scanning instrument with the ease of operation and frequency versatility you've dreamed of. Imagine, selecting from any of the public service bands and from all local frequencies by simply pushing a few buttons. No longer are you limited by crystals to a given band and set of frequencies. It's all made possible by *Bearcat* spaceage solid state circuitry. You can forget crystals forever.

Pick the 10 frequencies you want to scan and punch them in on the keyboard. It's incredibly easy. The large decimal display reads out each frequency you've selected. When you want to change frequencies, just enter the new ones.

Automatic search lets you scan any given range of frequencies of your choice within a band. Push-button lockout permits you to selectively skip frequencies not of current interest. The decimal display with its exclusive "rolling zeros" tells you which channels you're monitoring. When the *Bearcat* 210 locks in on an active frequency the decimal display shows the channel and frequency being monitored.

With the patented track-tuning system, the Bearcat 210 automatically aligns itself so that circuits are always "peaked" for any broadcast. Most competitive models peak only at the center of each band, missing the frequencies at the extreme ends of the band.

The Bearcat 210's electronically switched antenna eliminates the need for the long low band antenna. And a quartz crystal filter rejects adjacent stations as well as noise interference.

BankAmericard and Mastercharge cardholders can order on our 24-hour toll-free credit card order line 800-521-4414. In Michigan and outside the U.S. call 313-994-4441. Add $5 for U.S. shipping or $8 for air UPS to west coast. Charge cards or money orders only please. Foreign orders invited. Michigan residents add tax. For quantity pricing or if you have any questions, write: Communications Electronics, Engineering Division, P.O.Box 1002, Ann Arbor, Michigan 48106.

Bearcat® 210 Features

- **Crystal-less**—Without ever buying a crystal you can select from all local frequencies by simply pushing a few buttons.
- **Decimal Display**—See frequency and channel number—no guessing who's on the air.
- **5-Band Coverage**—Includes Low, High, UHF and UHF "T" public service bands, the 2-meter amateur (Ham) band, plus other UHF frequencies.
- **Deluxe Keyboard**—Makes frequency selection as easy as using a push-button phone. Lets you enter and change frequencies easily . . . try everything there is to hear.
- **Patented Track Tuning**—Receive frequencies across the full band without adjustment. Circuitry is automatically aligned to each frequency monitored.
- **Automatic Search**—Seek and find new, exciting frequencies.
- **Selective Scan Delay**—Adds a two second delay to prevent missing transmissions when "calls" and "answers" are on the same frequency.
- **Automatic Lock-Out**—Locks out channels and "skips" frequencies not of current interest.
- **Simple Programming**—Simply punch in on the keyboard the frequency you wish to monitor.
- **Space Age Circuitry**—Custom integrated circuits . . . a *Bearcat* tradition.
- **UL Listed/FCC Certified**—Assures quality design and manufacture.
- **Rolling Zeros**—This *Bearcat* exclusive tells you which channels your scanner is monitoring.
- **Tone By-Pass**—Scanning is not interrupted by mobile telephone tone signal.
- **Manual Scan Control**—Scan all 10 channels at your own pace.
- **3-Inch Speaker**—Front mounted speaker for more sound with less distortion.
- **Squelch**—Allows user to effectively block out unwanted noise.
- **AC/DC**—Operates at home or in the car.

Bearcat® 210 Specifications

Frequency Reception Range

Low Band	32—50 MHz
"Ham" Band	146—148 MHz
High Band	148—174 MHz
UHF Band	450—470 MHz
"T" Band	470—512 MHz

*Also receives UHF from 416—450 MHz

Size
10⅝" W x 3" H x 7⅝" D

Weight
4 lbs. 8 oz.

Power Requirements
117V ac, 11W; 13.8 Vdc, 6W

Audio Output
2W rms

Antenna
Telescoping (supplied)

Sensitivity
0.6μv for 12 dB SINAD on L & H bands
U bands slightly less

Selectivity
Better than −60 dB @ ± 25 KHz

Scan Rate
20 channels per second

Connectors
External antenna and speaker; AC & DC power

Accessories
Mounting bracket and hardware
DC cord

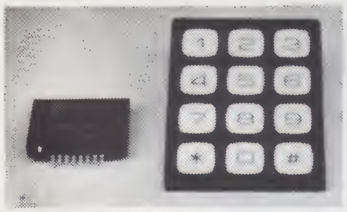
COMMUNICATIONS ELECTRONICS™

Box 1002 Dept. K-13
Ann Arbor, Michigan 48106 USA

master charge
THE INTERBANK CARD

BANKAMERICARD
welcome here

C5



TOUCH TONE ENCODER KIT

Simplicity itself to complete. No other parts required, no crystal required. The back of the touch pad has etched & drilled PC board and you solder the encoder chip to it. Add your own small speaker & 9 volt battery and you are done. A touch of the pad produces the proper tone signal from the speaker. We furnish schematic and instructions.

SP-149-B $12.95

WIRE WRAP WIRE

TEFZEL blue #30 Reg. price $13.28/100 ft. Our price 100 ft $2.00; 500 ft $7.50.

MULTI COLORED SPECTRA WIRE

		Footage	10′	50′	100′
8	Cond.	#24	$2.50	9.00	15.00
12	"	22	3.00	11.00	18.00
14	"	22	3.50	13.00	21.00
24	"	24	5.00	20.00	30.00
29	"	22	7.50	28.00	45.00

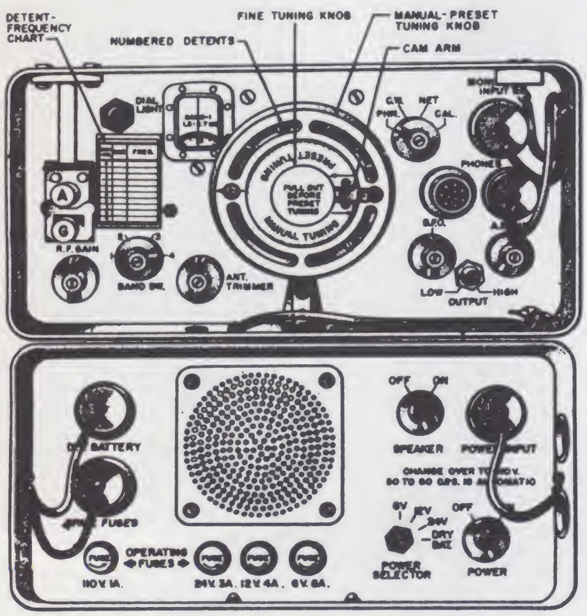
Great savings as these are about 1/4 book prices. All fresh & new.

TOUCHTONE ENCODER CHIP

Compatible with Bell system, no crystal required. Ideal for repeaters & w/specs. $6.00

CHARACTER GENERATOR CHIP

Memory is 512x5 produces 64 five by seven ASCII characters. New material w/data. $6.00



SHORT WAVE RECEIVER

Latest release by the Military. Here is a really useable piece of equipment requiring no conversion to use. Tuneable short wave radio receiver from 1.5 mc thru 18 mc with continuous tuning and ten pre-set detent channels. Great short wave listening. Replaces the older WW-2 BC-348 and TCS sets. Crystal calibration built-in, 200 KC steps. Comes with power supply to run on regular house current or from 6-12-24 volts DC. Built in speaker and amplifier. Brings in all Ham Bands from 1.5 to 18 Mc. Foreign broadcast, Italy, Germany, BBC Voice of America, even Russia. A Super Deal for the short wave listener. BFO also. We can ship in 2 packages by UPS or one package via truck. Shipping wgt total of 61 lbs. AM sensitivity 5 uv, 455 KC "IF" Sold "as is" visually inspected by us prior to shipping. Schematic included.

Ship wgt. 62 lbs. SPL-18-L $55.00

Singer high speed Dot Matrix Printer

New in cartons, similar to Centronics, prints up to six (6) copies, 16" width, 5 x 7 Dot matrix, 132 columns. No electronics or interface. Data supplied $300.00

IR NIGHT VIEWER $199.00

Custom made, complete with light source & viewer in one piece. Comes with carrying strap. Ready to operate with 6 volt lantern battery. Guaranteed by the manufacturer. See in total darkness. Great for scientists, viewing nocturnal animals & birds, criminal investigation . . . observe without being observed, and a ball for just plain snooping!!!! Sorry to say but no shipments to Calif. (lens may vary slightly from pic)

SPL-21 $199.00

Please add shipping cost on above. Minimum order $10

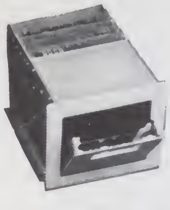
FREE CATALOG NOW READY # SP-10

Meshna

P.O. Box 62, E. Lynn, Massachusetts 01904

M2

PARALLEL I/O INCREMENTAL CASSETTE RECORDER



The PI-70 is a general purpose, parallel input and output, incremented and block digital recorder and reproducer. The standard Phillips cassette is the storage medium. The unit is made by International Computer Products Inc. of Dallas, Texas, and the selling price new is about $485.00.

The recorder employs a special recording technique, allowing the interchange of bytes up to 9 bits on each transfer. Data transfer may be made at variable rates, due to 4 possible operating speeds. Record timing, data encoding/decoding, motor controls are all contained internally.

All controls are electrically operable from the interface. Power to the recorder must be supplied from an external source. The PI-70 is 6¾" wide, 8" long, and 6½" deep, and weighs 4 lbs. Cassette access is provided by either an Amphenol or Cinch edge connector.

Available operating modes are: A. Incremental write forward, B. Incremental read forward, C. Incremental read reverse, D. Continuous write forward, E. Continuous read forward, F. Continuous read reverse, G. Rewind, H. Write forward from clear leader, I. Read forward from clear leader, and J. Write erase reverse. All input/output logic levels are DTL/TTL compatible.

Power required by the unit is as follows:

+12 VDC 0.7 A. 3% Reg.	Data rate:	Fast tape access: (bi-directional)
-12 VDC 0.7 A. 3% Reg.	Incremental, 0-50 bytes/sec.	Incremental format 400+ bytes/sec.
+5 VDC 1 A. 5% Reg.	Continuous, 40-80 bytes/sec.	Continuous format, 800+ bytes/sec.

Rewind: 45 sec. per 300 ft. Error rate: Less than 1 error per million characters, with approved tape.

We supply data and circuit diagrams. 100-page Technical Manual available from manufacturer.

STOCK No. 5564K $99.50

COMPUTER VIDEO MONITOR

A few months ago we advertised a limited quantity of 12" VIDEO MONITORS. We sold out in 2 days, as many of you know. We have now acquired another lot of 12" VIDEO MONITORS. These monitors are all in warranty returns to a large computer manufacturer. Rather than service these units, he replaced them with new units, and we acquired the lot. We have gone over them thoroughly, and guarantee them to be in working condition. We provide a 14-page service manual and wiring diagram.

Some of the specifications are as follows: Signal input, Composite-video & sync, per RS 170, 1 to 2 volts p-p. sync polarity negative. Separate Vert. & Hor. Sync, 1 to 5 volts positive, can be changed to negative. Impedance, equivalent to 150K ohms, parallel with 15 pf. 75 ohm terminated. Video response, within 3 dB from 15 Hz to 15 MHz. Resolution, center, 750, corner 650. Linearity, 1%. Picture Scan, 15,750 horiz. lines/sec. 47 to 63 Vert. pulses/sec. Horizontal retrace time 11 usec. max. Vertical retrace time, allow 21 scan lines per RS 170.

These monitors sold in excess of $250.00 each in quantities of 100. An opportunity to get a quality monitor at surplus prices.

STOCK No. 5525K VIDEO MONITOR, tested and working, with data Wt. 20 lbs. $89.50

3 NEW BLOCKBUSTER TRANSFORMERS

TRANSFORMER 1. 115 primary, Secondary 1, 30 V @2A. Secondary 2, 16.5 V @1.2A, Secondary 3, 16 V @3.5A, Secondary 4, 9.5 V @3.5A. STOCK NO. 6667K 10 lbs. $10.95 2/$20.00

TRANSFORMER 2. Duall primary. Secondary 1, 12V @5A. Secondary 2, 24 V @9A. Secondary 3, 14 V @20A. Secondary 4, 125V @1.5A. Wt. 16 lbs. STOCK No. 6675N $18.95 ea. 2/$36.00

TRANSFORMER 3, 2 different primaries. Following 3 voltages & currents with primary 1. Secondary 1, 9.8 V @8.8A. Secondary 2.18 volts @5A. Secondary 3, 21 V @6.5A. Primary 2, secondary 1, 5.8 V @8.8A, Secondary 2, 10 V @ 5A. Secondary 3, 12 V @6.5A. Wt. 10 lbs. STOCK No. 6675K $12.95 2/$24.00

WIRE WRAP PROTOTYPE BOARDS

Wire wrap is the thing today, whether you are adding memory to your computer, building from scratch, designing new circuits, etc. We have 4 boards, 2 out of equipment, that have wire on them that must be removed (easy with an OK wire wrap tool), and 2 virgin boards. Board 6558K has from 75 to 100 sockets 14 & 16 pin. Board 6559K has from 40 to 50 sockets, 14 & 16 pin. Board 6592K has 40 16-pin sockets, and 4 LSI sockets, and is gold-plated. All pins are brought up to the top of this board for ease in wiring. Board 5561K has 87½ sockets, 28 16-pin sockets and a 4-pin socket. Has heavily by-passed Vcc and ground planes.

STOCK No. 6558K	75 to 100 sockets	5¾x13¾	$18.75 ea.	2/$36.00
STOCK No. 6559K	40 to 50 sockets	6"x6½"	$11.75 ea.	2/$22.00
STOCK No. 6592K	40 16-pin sockets 4 LSI sockets	6¾x8¾"	$24.50 ea.	2/$45.00
STOCK No. 5561K	88½ sockets	4½"x14½"	$29.50 ea.	2/$55.00

DELTA ELECTRONICS
7 Oakland St.
Amesbury, Mass. 01913
(617) 388-4705 D13

master charge THE INTERBANK CARD · Your BANKAMERICARD welcome here

ATLANTA, GEORGIA
DELTA ELECTRONIC HOBBIES, 5151 Buford Hwy., Doraville, Atlanta, Ga. Retail only.

BOSTON, MASSACHUSETTS
DELTA ELECTRONICS WAREHOUSE OUTLET, 590 Commonwealth Ave., Boston, Ma. Retail only.

RONDURE COMPANY

2522 BUTLER ST. • DALLAS, TEXAS 75235 • 214-630-4621

the computer room R7

Specifications
- Size: 21" wide x 21" deep x 8" high
- Power Input 115 Volt 60 Hz
- Interface: RS232
- Weight: 54 lbs. (Shipping Weight 65 lbs.)
- 15" Carriage
- Input/Output rates to 15 characters per second
- EBCD Code
- Half Duplex
- 132 Print Positions, 10 Pitch
- Can be used off-line

Used
Working
(Non Refurbed) $695.

SPECIAL $650.00

Software to connect ASCII Output of 8080 Class Processor to Selectric: Code $25
Manufacturers Electronic & Mechanical Documentation
$20. with machine $40. Documentation only

SELECTRIC TERMINAL (IBM Selectric Mechanism, Heavy Duty, Datel Electronics)

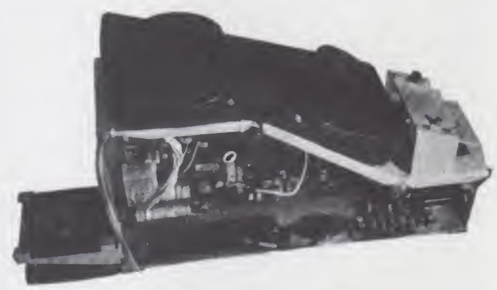
CARTERFONE MODEL 318 ASYNCH MODEM

- HARD WIRE
- TTY OR RS-232B INTERFACE
- ORIGINATE ONLY
- UP TO 300 BPS

USED – UNTESTED . . $25.00
USED – TESTED $80.00

We ship prints with these.

KEYBOARDS
Used ASCII Keyboards with enclosure & documentation
$55.00



NOVATION ACOUSTICAL MODEM
Originate Only
Used – Untested
Physically fit into Model 33 Teletype.
Manufactured by Novation
$25 each
(Order item #M-7)

IC'S

IC's (New, Tested)

74H00	–	7 for $1.00
74H04	–	7 for $1.00
7410	–	10 for $1.00
7430	–	7 for $1.00
7440	–	10 for $1.00
74170	–	2 for $1.00
74195	–	3 for $1.00
74H60	–	2 for $1.00

CINCH EDGE CONNECTOR
(NEW) (Dual 22) $3.00 each

DUAL 30 W W 2.00 each

SUPPLIES TTY Paper – $2.00/Roll
TTY Tape – $2.00/Roll

USED RIBBON
Cable
Assortment

2 lb. Bundle $ 5.00
5 lb. Bundle $10.00

SUGART MINI-FLOPPY DRIVE

NEW PRICE
$355.00 Each
~~$390.00 each~~

MODEL
SA-400

SHIPPING INFORMATION:

Modems: $2.00 each; 2 for $4.00 UPS
Small Items & Parts: $2.00/order less than $20.00; $4.00/order $20.00 to $100.00; $6.00/order over $100.00
Large Items & Parts: Specify Freight or Air Freight Collect
Foreign Orders: Add appropriate freight or postage
Please specify exactly what you wish by order number or name or both.
We now take Master Charge orders. Specify full number. bank number and expiration date.

ORDERING INFORMATION:

In general no cords or cables are shipped unless we specify that they are supplied.
We ship the same day we receive a certified check or money order.
Texas residents add 5% sales tax.
Please call if you have a question.
All items subject to availability. Your money returned if we are out of stock.
Items are either new (specified) or they are used (tested or untested) and no other warranty is made or implied.

DIODES/ZENERS

1N914	100v	10mA	.05
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.05
1N753A	6.2v	z	.25
1N758A	10v	z	.25
1N759A	12v	z	.25
1N4733	5.1v	z	.25
1N5243	13v	z	.25
1N5244B	14v	z	.25
1N5245B	15v	z	.25

SOCKETS/BRIDGES

8-pin	pcb	.25	ww	.45
14-pin	pcb	.25	ww	.40
16-pin	pcb	.25	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.45	ww	1.25
24-pin	pcb	.35	ww	1.10
28-pin	pcb	.35	ww	1.45
40-pin	pcb	.50	ww	1.25

Molex pins	.01	To-3 Sockets	.45
2 Amp Bridge		100-prv	1.20
25 Amp Bridge		200-prv	1.95

TRANSISTORS, LEDS, etc.

2N2222	NPN	(Plastic .10)	.15
2N2907	PNP		.15
2N3906	PNP		.10
2N3054	NPN		.35
2N3055	NPN	15A 60v	.50
T1P125	PNP	Darlington	.35
LED Green, Red, Clear			.15
D.L.747	7 seg 5/8" high com-anode		1.95
XAN72	7 seg com-anode		1.50
FND 359	Red 7 seg com-cathode		1.25

C MOS

4000	.15
4001	.20
4002	.20
4004	3.95
4006	1.20
4007	.35
4008	.95
4009	.30
4010	.45
4011	.20
4012	.20
4013	.40
4014	1.10
4015	.95
4016	.35
4017	1.10
4018	1.10
4019	.60
4020	.85
4021	1.35
4022	.95
4023	.25
4024	.75
4025	.35
4026	1.95
4027	.50
4028	.95
4030	.35
4033	1.50
4034	2.45
4035	1.25
4040	1.35
4041	.69
4042	.95
4043	.95
4044	.95
4046	1.75
4049	.70
4050	.50
4066	.95
4069	.40
4071	.35
4081	.70
4082	.45

— T T L —

7400	.15
7401	.15
7402	.20
7403	.20
7404	.15
7405	.25
7406	.35
7407	.55
7408	.25
7409	.15
7410	.10
7411	.25
7412	.30
7413	.45
7414	1.10
7416	.25
7417	.40
7420	.15
7426	.30
7427	.45
7430	.15
7432	.30
7437	.35
7438	.35
7440	.25
7441	1.15
7442	.45
7443	.85
7444	.45
7445	.65
7446	.95
7447	.95
7448	.70
7450	.25
7451	.25
7453	.20
7454	.25
7460	.40
7470	.45
7472	.40

7473	.25
7474	.35
7475	.35
7476	.30
7480	.55
7481	.75
7483	.95
7485	.95
7486	.30
7489	1.35
7490	.55
7491	.95
7492	.95
7493	.40
7494	1.25
7495	.60
7496	.80
74100	1.85
74107	.35
74121	.35
74122	.55
74123	.55
74125	.45
74126	.35
74132	1.35
74141	1.00
74150	.85
74151	.75
74153	.95
74154	1.05
74156	.95
74157	.65
74161	.85
74163	.95
74164	.60
74165	1.50
74166	1.35
74175	.80

74176	1.25
74180	.85
74181	2.25
74182	.95
74190	1.75
74191	1.35
74192	1.65
74193	.85
74194	1.25
74195	.95
74196	1.25
74197	1.25
74198	2.35
74221	1.00
74367	.85
75108A	.35
75110	.35
75491	.50
75492	.50
74H00	.25
74H01	.25
74H04	.25
74H05	.25
74H08	.35
74H10	.35
74H11	.25
74H15	.30
74H20	.30
74H21	.25
74H22	.40
74H30	.25
74H40	.25
74H50	.25
74H51	.25
74H52	.15
74H53J	.25
74H55	.25

74H72	.55
74H101	.75
74H103	.75
74H106	.95
74L00	.35
74L02	.35
74L03	.30
74L04	.35
74L10	.35
74L20	.35
74L30	.45
74L47	1.95
74L51	.45
74L55	.65
74L72	.45
74L73	.40
74L74	.45
74L75	.55
74L93	.55
74L123	.55
74S00	.55
74S02	.55
74S03	.30
74S04	.35
74S05	.35
74S08	.35
74S10	.35
74S11	.35
74S20	.35
74S40	.25
74S50	.25
74S51	.45
74S64	.25
74S74	.40
74S112	.90
74S114	1.30

74S133	.45
74S140	.75
74S151	.35
74S153	.35
74S157	.80
74S158	.35
74S194	1.05
74S257 (8123)	.25
74LS00	.35
74LS01	.35
74LS02	.35
74LS04	.35
74LS05	.45
74LS08	.35
74LS09	.35
74LS10	.35
74LS11	.35
74LS20	.35
74LS21	.25
74LS22	.25
74LS32	.40
74LS37	.35
74LS40	.45
74LS42	1.10
74LS51	.50
74LS74	.65
74LS86	.65
74LS90	.95
74LS93	.95
74LS107	.85
74LS123	1.00
74LS151	.95
74LS153	1.20
74LS157	.85
74LS164	1.90
74LS367	.85
74LS368	.85

9000 SERIES

9301	.85
9309	.35
9322	.85
95H03	.55
9601	.75
9602	.50

MEMORY CLOCKS

74S188 (8223)	3.00
1702A	6.95
MM5314	3.00
MM5316	3.50
2102-1	1.75
2102L-1	1.95
TR 1602B/ TMS 6011	6.95
8080AD	15.00
8T13	1.50
8T23	1.50
8T24	2.00
2107B-4	4.95

LINEARS, REGULATORS, etc.

8266	.35
MCT2	.95
8038	3.95
LM201	.75
LM301	.25
LM308 (Mini)	.75
LM309H	.65
LM309K (340K-5)	.85
LM310	1.15
LM311D (Mini)	.75
LM318 (Mini)	.65

LM320K5 (7905)	1.65
LM320K12	1.65
LM320T5	1.65
LM320T12	1.65
LM320T15	1.65
LM339	.95
7805 (340T5)	.95
LM340T12	1.00
LM340T15	1.00
LM340T18	1.00

LM340T24	.95
LM340K12	2.15
LM340K15	1.25
LM340K18	1.25
LM340K24	.95
LM373	2.95
LM380	.95
LM709 (8,14 PIN)	.25
LM711	.45

LM723	.50
LM725	1.75
LM739	1.50
LM741 (8-14)	.25
LM747	1.10
LM1307	1.25
LM1458	.95
LM3900	.50
LM75451	.65
NE555	.50
NE556	.95
NE565	.95
NE566	1.75
NE567	1.35

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Boulevard, San Diego, California 92111
(714) 278-4394 *(Calif. Res.)*

All orders shipped prepaid — No minimum
Open accounts invited — COD orders accepted

Discounts available at OEM Quantities — California Residents add 6% Sales Tax
All IC's Prime/Guaranteed. All orders shipped same day received.

24 Hour Toll Free Phone 1-800-854-2211 — **MasterCharge / BankAmericard / AE**

SPECIAL DISCOUNTS

Total Order	Deduct
$35 - $99	5%
$100 - $300	10%
$301 - $1000	15%
$1000 - Up	20%

MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of *Kilobaud*.
Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly. Write to:

• KILOBAUD • PETERBOROUGH NH 03458 •

☐ Address change only
☐ Extend subscription
☐ Enter new subscription
☐ 1 year $15.00
☐ Payment enclosed (1 extra BONUS issue)
☐ Bill me later

Name ______________________ Call ________
Address ______________________________
City __________ State ________ Zip ________

If you have no label handy, print OLD address here.

AFFIX LABEL

Name ______________________ Call ________
Address ______________________________
City __________ State ________ Zip ________

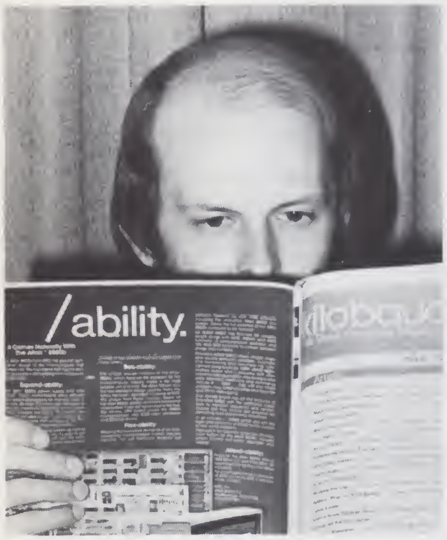
Who Reads kilobaud?

The staff of Kilobaud magazine would like to know more about you! As a Christmas present to us, would you please fill out the following questionnaire and return it by December 31, 1977.
Kilobaud • Peterborough NH 03458

demographic profile

City ______________________ State ________
Age ________ Sex ________
Education ______________________________
____
Profession ______________________________
Do you own a system now? ________ What kind? ________________
____
Present equipment budget (hardware/software)? ________________
Projected equipment budget (hardware/software) for 1978? ________________

KB/1/78



Answer next month

WHO'S BEHIND KILOBAUD?

What's in a name?

A lot, when it's KILOBAUD! *(the high capacity data channel for computerists)* and you can get a lot of

kilobaud

for only $15 a year

Save $9 over single copy price.

The most interesting and understandable computer magazine written for beginners *(experts prefer it too!)* should be coming to your very own mailbox every month.

SUBSCRIBE NOW!

kilobaud

PETERBOROUGH NH 03458

YES! I'LL SUBSCRIBE!

Send me Kilobaud for:

☐ 1 year — $15.00* ☐ Renewal

☐ 3 years — $36.00** ☐ New sub.

Enclosed $________ ☐ Cash ☐ Check ☐ Money Order

☐ American Express ☐ BankAmericard/Visa ☐ Master Charge

Credit card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiration date ________

☐ Bill me directly/sign below

Signature ______________________________

Address ______________________________

City ____________________ State ______ Zip ______

* U.S. & Canada only. Others write for foreign rates. KB/13/78

**U.S. & Canada only. TOLL FREE SUBSCRIPTION NUMBER: 1-800-258-5473

what do you give the man who has everything?

a box to put it in.

Kilobaud, as thick as it is, is more like a floppy when it comes to standing on the bookshelf. Enter the new Kilobaud Library Shelf Boxes, sturdy corrugated cardboard boxes which will hold your magazines on the shelf and keep them from flopping around.

Yes, we know all about binders . . . we have them too . . . and we sell them, but binders are a drag when you want one copy of a magazine. And they cost like sin (which costs plenty).

Just to be rotten (a talent we are trying to develop, but which comes hard), we have self-sticking labels for the boxes, not only for *Kilobaud*, but also for *73 Magazine* . . . and for *Personal Computing, Radio Electronics, Popular Electronics, Interface Age*, and . . . yep . . . *Byte*. Heh, heh! Just ask for whatever stickers you want and we'll throw 'em in with your box order. Hams may want our labels for *CQ, QST* or *Ham Radio*, if they get any of those magazines. This is a way you can buy one set of matching boxes and line 'em up on your shelf . . . looks very nice that way.

The boxes are a white color and are particularly resistant to dirt, a real plus for white boxes. There's some kind of funny plastic finish on 'em.

You'll probably do like most people who have tried these so far and order one or two for starters . . . then get a couple dozen. The postage on these is the killer . . . so one box costs $2.00 postpaid and each additional is $1.50 each postpaid.

Unless the magazine gets a whole lot fatter than it is right now these boxes should hold a full year of *Kilobaud* . . . or 73.

One side is cut low to permit you to see the binding of the magazine . . . and note that we are now printing the information on the top part of the binding so it will show in these boxes. You can put the boxes on your shelves with the bindings showing or with just the white board showing, there are little marks to help you center your labels on either side.

Your magazine library is your prime reference, so keep it handy and keep it neat with these strong library shelf boxes.

Send me ______ boxes for Kilobaud Shelf Storage at $2.00 for the first box and $1.50 for each additional box. Include the following labels: ______________________

____

$______ Enclosed ☐ Cash ☐ Check ☐ M.O.
Bill: ☐ American Express ☐ BankAmericard ☐ Master Charge

Interbank # ☐☐☐☐ Expiration date ______

Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature ______________________

Name ______________________

Address ______________________

City ______________ State ______ Zip ______

KILOBAUD LIBRARY SHELF BOXES
Peterborough NH 03458
or call Toll Free (800) 258-5473

KB/1/78

COMPUTER MAILING LIST

Completely re-written and updated to include dealers, clubs, publications, and manufacturers. (It's the one we use for our mailings and we update it daily.) The list has over 1700 names painstakingly gathered from manufacturers, magazine ads, new product releases, hobby computer shows, and direct mail. You can buy this list printed on self-sticking labels for only $50.

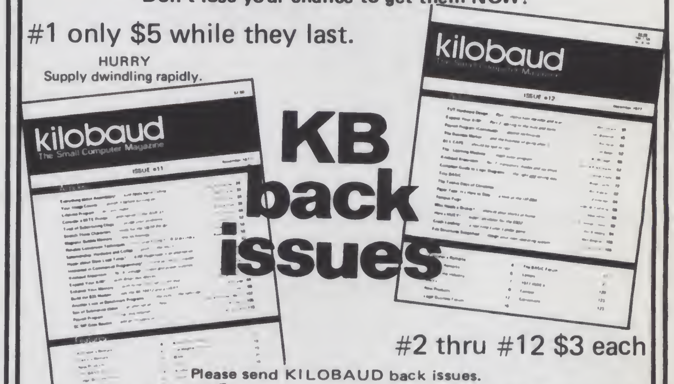
kilobaud

PETERBOROUGH NH 03458

Did you miss out on the first issue of Kilobaud?
Don't lose your chance to get them NOW!

#1 only $5 while they last.

HURRY
Supply dwindling rapidly.



KB back issues

#2 thru #12 $3 each

Please send KILOBAUD back issues.

___ issues JAN 77 $5	___ issues MAY 77 $3	___ issues SEPT 77 $3
___ issues FEB 77 $3	___ issues JUNE 77 $3	___ issues OCT 77 $3
___ issues MAR 77 $3	___ issues JUL 77 $3	___ issues NOV 77 $3
___ issues APR 77 $3	___ issues AUG 77 $3	___ issues DEC 77 $3

_____ TOTAL

☐ BankAmericard ☐ Master Charge ☐ American Express

Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Interbank # ☐☐☐☐

Expiration date __________

Signature ______________________

Name ______________________

Address ______________________

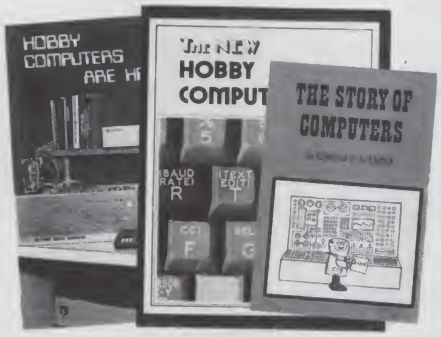
City ______________ State ______ Zip ______

Have credit card # handy and call
TOLL FREE (800) 258-5473
KILOBAUD • PETERBOROUGH NH 03458

KB/1/78

Kb book nook Kb book nook

beginner's·introductory

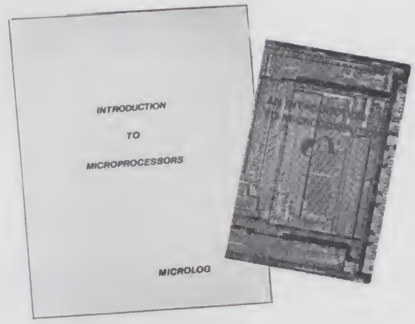


●**THE STORY OF COMPUTERS** by Donald D. Spencer is to computer books what *Dick and Jane* is to novels . . . extremely elementary, gives the non-computerist a fair idea of what the hobbyist is talking about when he speaks computer lingo. Attempts to explain what computers are and can do to a spouse, child or any un-electronics-minded friend. $4.95.*

●**HOBBY COMPUTERS ARE HERE** If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

●**THE NEW HOBBY COMPUTERS!** This book takes it from where "Hobby Computers Are Here" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! $4.95.*

●**HOME COMPUTERS: 2^{10} Questions & Answers** by Rich Didday. Two books aimed exclusively at the novice computer hobbyist/home computer user. Written in a rather unusual style which has a beginner asking questions which are answered by a person with a substantial background in computers and personal computing. The questions are just the kind beginners come up with . . . and the answers are presented in easy-to-understand terms (usually with a diagram to illustrate the point). Both the hardware and software aspects of home computing are covered from A to Z. An index in both books makes them ideal as reference material *for anyone*. Volume I: Hardware — $7.95*; Volume 2: Software — $6.95*.

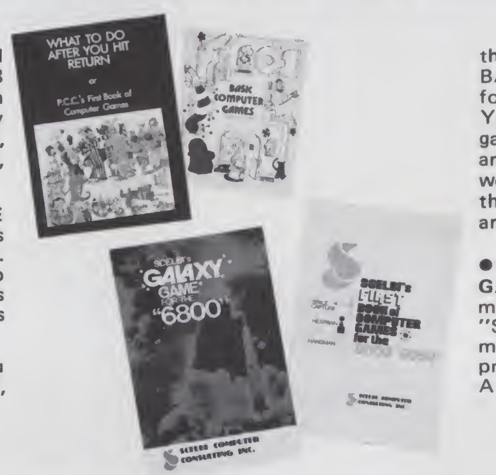


●**MICROCOMPUTER PRIMER** by Mitchell Waite and Michael Pardee. Describes basic computer theory, explains numbering systems, and introduces the reader to the world of programming. Describes the world of microcomputing in "real world" terminology. There is probably no better way of getting involved with the exciting new hobby of microcomputing. $7.95.*

●**INTRODUCTION TO MICROPROCESSORS** by Charles Rockwell of MICROLOG is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Addressing schemes, registers, control, and memory are all explained. General information about hardware systems is provided. Specific systems are not described and programming is only briefly discussed. $17.50 US and Canada, $20 elsewhere.*

●**AN INTRODUCTION TO MICROCOMPUTERS, VOLS. 1 AND 2** by Adam Osborne Associates, are references dealing with microcomputer architecture in general and specifically with details about most of the common chips. These books are not software-oriented, but are invaluable for the hobbyist who is into building his own interfaces and processors. Volume 1 is dedicated to general hardware theory related to micros, and Volume 2 discusses the practical details of each micro chip. (Detailed review in Kilobaud #2) Volume 1 — $7.50*; Volume II — $12.50.*

computer games



●**WHAT TO DO AFTER YOU HIT RETURN** PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, muchly illustrated. Lunar landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00.*

●**SCELBI'S GALAXY GAME FOR THE "6800"** Here's a new twist in computer games by Robert Findley/Raymond Edwards. "Galaxy" pits the operator of a spaceship against alien craft, as well as such variables as speed, time, and ammunition. No two games are the same! $14.95.*

●**101 GAMES IN BASIC** Okay so once you get your computer up and running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

●**SCELBI'S FIRST BOOK OF COMPUTER GAMES** Need a game for your 8008 or 8080 microprocessor? Has three popular games, "Space Capture," "Hexpawn," and "Hangman." Complete flowcharts, logic description, program listing, and instructions are provided. A must for the game freak! $14.95.*

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department Peterborough NH 03458
Be sure to include check or detailed credit card information
* *Add $1.00 shipping & handling charge for each order.* Note: Prices subject to change on books not published by 73 Magazine.

software·programming

●**BASIC** by Bob Albrecht, etc. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $4.95.*

●**A QUICK LOOK AT BASIC** by Donald D. Spencer. A perfect reference for the beginning programmer. Assumes that the reader has no previous programming experience and is a self-teaching guide for the individual desiring to learn the fundamentals of BASIC, the most common hobbyist programming language. $4.95.*

●**MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC** An introduction to BASIC . . . simple enough for your kids. If you want to teach BASIC to anyone quickly, this book is the way to go. $2.00.*

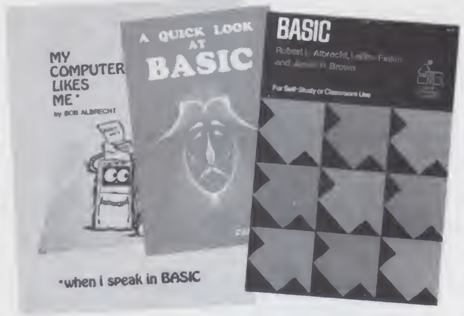
●**FUN WITH COMPUTERS AND BASIC** by Donald D. Spencer, contains an easy-to-understand explanation of the BASIC Programming Language and is intended for persons who have had no previous exposure to computer programming but want to learn BASIC quickly, easily, and interestingly. Over half the book is devoted to problems using games, puzzles, and mathematical recreations (you don't need a math background to understand most of the problems in this book). A superior book for self-teaching and learning computer programming. $6.95.*

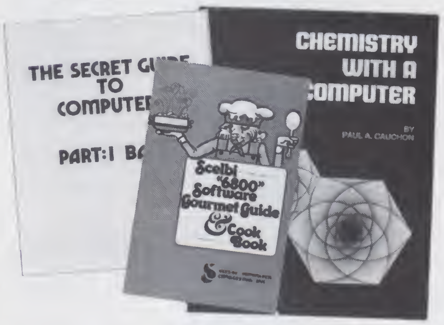
●**SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS** by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Incudes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

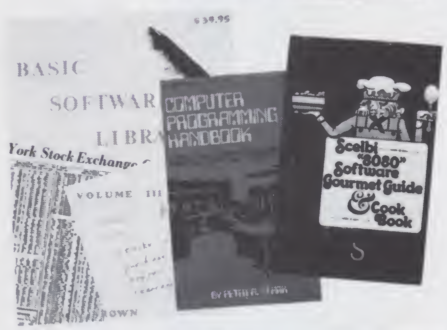
●**THE SECRET GUIDE TO COMPUTERS** Parts 1, 2, and 3 by Russ Walter. **Part One** describes computers in general, and after reading for ten minutes you will be writing simple BASIC programs! Employs a step-by-step teaching process, the end result being a working knowledge of BASIC. **Part Two** discusses computer applications. It's one thing to master the syntax of a language such as BASIC and another to solve problems using the new tool. This also provides useful techniques in problem solving. **Part Three** describes programming languages. Ever heard of APL and QLISP? BASIC is not the only language used to program computers, and as the hobby industry grows additional languages will become commonplace. 7th Edition. Part I — $2.75*; Part II — $2.50*; Part III — $3.50.*

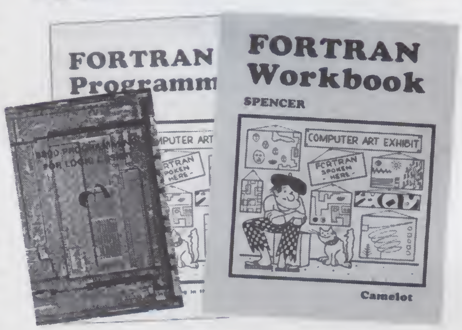
●**SOME COMMON BASIC PROGRAMS** published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs like depreciation formulas, loan payment solutions and loan interest rates; math routines containing statistical packages and polynomial derivations . . . plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $7.50 paperback.*

●**Scientific Research Instruments' BASIC SOFTWARE LIBRARY** is a complete do-it-yourself kit. Written in everybody's BASIC immediately executable in ANY computer with at least 4K, no other peripherals needed. Over 700 pages of source codes, descriptions and instructions. Vol. I contains business and recreational programs and is 300 pages. Vol. II is 260 pages and contains math, engineering, statistics and plotting programs. Vol. III contains money managing, advanced business programs such as billing, A/R, inventory, payroll, etc. Vol. IV contains general purpose programs like loans, rates, retirement, plus games: Poker, Enterprise (take charge while Capt. Kirk is away), Football and more! Vol. V is filled with experimenter's programs including games, pictures and misc. problems like "logic." Vols. I & II $24.95,* Vol. III $39.95,* Vol. IV & V $9.95 each.*









●**COMPUTER PROGRAMMING HANDBOOK** by Peter Stark. A complete guide to computer programming and data processing. Includes many worked out examples and history of computers. $8.95.*

●**8080 PROGRAMMING FOR LOGIC DESIGN** Ideal reference for an in-depth understanding of the 8080 processor. The work is application-oriented, and the 8080 is discussed in light of replacing conventional, hard-wired logic. Both hardware and software are described. Practical design considerations are provided for the implementation of an 8080-based control system. (Detailed review Kilobaud #1) Pub. by Osborne Assoc. $7.50.*

●**8080 SOFTWARE GOURMET GUIDE AND COOKBOOK** If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. This manual describes sorting, searching, and many other routines for the 8080 user. $9.95.*

●**6800 PROGRAMMING FOR LOGIC DESIGN** Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Hardware, software and interfacing techniques are discussed. This reference, and its companion dedicated to 8080 users, provide practical information that allows an experimenter to design a complete micro control system from the "ground up." An excellent reference! Published by Osborne Associates, $7.50.*

●**6800 SOFTWARE GOURMET GUIDE & COOKBOOK** If you have been spending too much time developing routines for your 6800 microprocessor, try the new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other necessary routines for the 6800 user. $9.95.*

●**FORTRAN PROGRAMMING** by Donald Spencer. If you are familiar with BASIC you will appreciate the additional capabilities of FORTRAN, a computer language with most of BASIC's features, and much more! FORTRAN was designed for complex numeric calculations; and possesses extended I/O capability. It is easily learned, as it is an English-like computer language. $7.50.*

●**FORTRAN WORKBOOK** by Donald Spencer. Provides practical examples and problems to solve. Flowcharting is also discussed. Almost all micros support BASIC — it won't be long before FORTRAN is commonplace. Don't miss the programming boat . . . learn FORTRAN; and be ready for the next language boom! $3.95.*

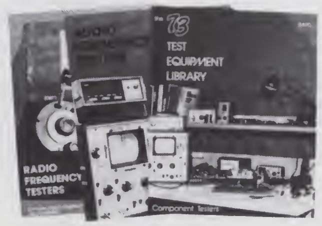
●**CHEMISTRY WITH A COMPUTER** by Paul A. Cauchon, contains a collection of tutorial, simulation and problem-generation computer programs. Simulations provide models of lengthy laboratory experimentation beyond the limited classroom timeframe and enhancement of course studies by encouraging prelaboratory research. Usable with almost any chemistry course in the high school or college level. $9.95.*

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department Peterborough NH 03458
Be sure to include check or detailed credit card information

** Add $1.00 shipping & handling charge for each order.* Note: Prices subject to change on books not published by 73 Magazine.

KB book nook KB book nook

hardware



●**TTL COOKBOOK** by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. 336 pages; 5½ x 8½; softbound $8.95.*

●**CMOS COOKBOOK** by Don Lancaster. Another winner from the author of the famous *RTL* and *TTL Cookbooks.* Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Eight chapters cover all facets of CMOS logic, and is prefaced by 100 pages detailing the characteristics of most CMOS circuits. Required reading for every serious digital experimenter! $9.95.*

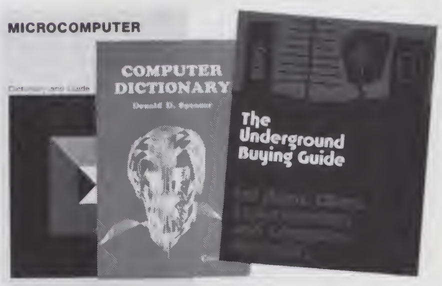
●**TVT COOKBOOK** by Donald Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. This book is a required text for both the microcomputer enthusiast and the amateur RTTY operator who desires a quiet alternative to noisy teletype machines. $9.95.*

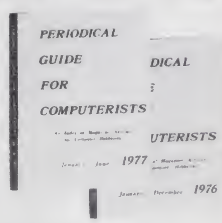
●**VOL I COMPONENT TESTERS** Build your own test equipment and save a bundle (and have a lot of fun). Volume I of the 73 Test Equipment Library shows you how to build and use transistor testers (8 of 'em), three diodes testers, 3 IC testers, 9 voltmeters and VTVMs, 8 ohmmeter, 3 inductance meters, and a raft of other gadgets for checking temperature, crystals, Q, etc. $4.95.*

●**VOL II AUDIO FREQUENCY TESTERS** If you're into audio . . . such as digital cassette recording, RTTY, Baudot vs ASCII, SSTV, SSB, Touchtone or even hi-fi . . . you'll want to have this book full of home built test equipment projects. Volume II $4.95.*

●**VOL III RADIO FREQUENCY TESTERS** This is of more interest to hams and CBers . . . test equipment you can build for checking out transmitters and receivers: signal generators, noise generators, crystal calibrators, GDOs, dummy loads . . . things like that. $4.95.*

general





●**COMPUTER DICTIONARY** by Donald D. Spencer. Defines words and acronyms used by computerists in a clear, easy to understand style. Over 2000 definitions are provided. This reference is a must for the individual getting started in the world of microcomputers. Published by Camelot Press, $5.95.*

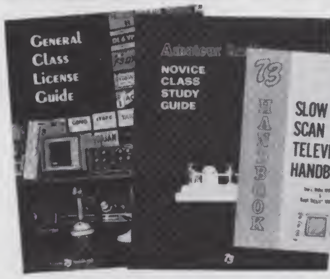
●**BRAND NEW DICTIONARY** This microcomputer dictionary fills the need to become quickly acquainted with the terminology and nomenclature of the revolution in computers. Over 5000 definitions and explanations of terms and concepts (704 pages) relating to microprocessors, microcomputers and microcontrollers. There is also a comprehensive electronics/computer abbreviations and acronyms section. $15.95.*

●**THE UNDERGROUND BUYING GUIDE** Here is a handy guide for the electronics enthusiast. Over 600 sources of equipment and literature are provided; some are mail-order-only outfits that do not advertise. Sources are listed alphabetically, by service or product, and by state. The guide is cross-referenced for ease of use. Electronic publishing houses are also listed. $5.95 each.*

●**1976 PERIODICAL GUIDE FOR COMPUTERISTS** is a 20-page book which indexes over 1,000 personal computing articles for the entire year of 1976 from Byte, Creative Computing, Digital Design, Dr. Dobbs Journal, EDN, Electronic Design, Electronics, Interface Age, Microtrek, Peoples Computer Company, Popular Electronics, QST, Radio Electronics, SCCS Interface and 73 Amateur Radio. Articles are indexed under more than 100 subject categories. Price $3.00.* New January — June 1977 Edition — $3.00.*

●**TYCHON'S 8080 OCTAL CODE CARD** Slide rule-like aid for programming and debugging 8080 software contains all the mnemonics and corresponding octal codes. Instructions are color coded to indicate which flags are affected during execution. Only 6½x3 inches provides a neat, logical format for quick reference. Reverse of card is printed with an ASCII code chart for 128 characters plus the 8080 status word and register pair codes. Also available, Tychon's *8080 Hex Code Card*, same as above only has hex codes instead of octal. $3.00 each.*

amateur radio books



●**NOVICE STUDY GUIDE** The most complete Novice study guide available. It is brand new. This is not only invaluable for anyone wanting to get started in amateur radio, but also it is about the only really simple book on the fundamentals of electricity and electronics. And without your fundamentals down pat, how can you go on to really understand and work with computers? First things first. $4.95.*

●**GENERAL CLASS STUDY GUIDE** Takes over on theory where the Novice book leaves off. You'll need to know the electronic theory in this to work with computers and you'll not find an easier place to get the information. It will also make getting your Tech or General license a breeze . . . then you can get on the ham repeaters and interconnect your micro with others. $5.95.*

●**SSTV HANDBOOK** This excellent book tells all about it, from its history and basics to the present state-of-the-art techniques. Contains chapters on circuits, monitors, cameras, color SSTV, test equipment and much more. Hardbound $7,* Softbound $5.*

●**VHF ANTENNA HANDBOOK** This new handbook details the theory, design and construction of hundreds of different VHF and UHF antennas . . . a practical book written for the average amateur who takes joy in building, not full of complex formulas for the design engineer. Packed with fabulous antenna projects you can build. $4.95.*

●**WEATHER SATELLITE HANDBOOK** Simple equipment and methods for getting good pictures from the weather satellite. Antennas, receivers, monitors, facsimile you can build, tracking, automatic control (you don't even have to be home). Dr. Taggart WB8DQT $4.95.*

●**THE NEW RTTY HANDBOOK** is a brand new 1977 edition and the only up-to-date RTTY book available. The state-of-the-art has been changing radically and has made all previous RTTY books obsolete. It has the latest circuits, great for the newcomer and the expert alike. $5.95.*

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department Peterborough NH 03458 Be sure to include check or detailed credit card information

* *Add $1.00 shipping & handling charge for each order.* Note: Prices subject to change on books not published by 73 Magazine.

kilobaud

index of advertisers

PETERBOROUGH NH 03458

kilobaud reader service

Circle appropriate Reader Service # for desired company brochures, data sheets or catalogs and mail to Kilobaud Magazine, Peterborough NH 03458. Include your zip code, please. Send money directly to advertisers. LIMIT: 25 requests.

A37	C53	C49	G4	J1	O5	R12	S28	V8
A38	C41	C34	H20	M17	O2	R14	S29	W13
A48	C36	C31	H5	M2	O3	R16	T11	W5
A42	C28	D13	I18	M27	P9	R7	T8	W16
A43	C40	D12	I13	M16	P7	S26	T26	KILOBAUD
B21	C33	D20	I9	M30	P12	S2	T13	
B14	C48	E21	I10	M28	P24	S5	T20	
C50	C44	F4	J6	M25	P21	S6	T1	

PLEASE PRINT OR TYPE ☐ Newsstand buyer ☐ Subscriber

This Month's ARTICLE WINNER: (Title page number) ____________

Name ____________

Address ____________

City ____________ State ________ Zip ________

Coupon Expires in 60 Days 1/78

PLACE STAMP HERE

kilobaud

PETERBOROUGH NH 03458

Att: Mail Order

I want to subscribe!

Send me Kilobaud for:

☐ 1 year — $15.00*

☐ 3 years — $36.00**

☐ Life Subscription — $155.00**

☐ Renewal

☐ New sub.

Enclosed $__________ ☐ Cash ☐ Check ☐ Money Order

☐ American Express ☐ BankAmericard ☐ Master Charge

Credit card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiration date ________ Signature ____________

☐ Bill me directly Signature ____________

Name ____________

Address ____________

City ____________ State ________ Zip ________

* U.S. & Canada only. Others write for foreign rates.

**U.S. & Canada only.

Subscription prices expire in 60 days. 1/78

FIRST CLASS
Permit No. 1024
Peterborough NH 03458

BUSINESS REPLY MAIL

NO POSTAGE NECESSARY IF MAILED IN UNITED STATES

POSTAGE WILL BE PAID BY

kilobaud

PETERBOROUGH NH 03458

Att: Reader Service

PLEASE PRINT OR TYPE   Please send me the following Kilobaud products:

Quantity	Description	Unit Price	Total

Total ______

Shipping & Handling Charge $1.00

Order Total ______

Enclosed $_____ ☐ Cash ☐ Check ☐ Money Order ☐ C.O.D.

Bill: ☐ American Express ☐ BankAmericard ☐ Master Charge

Credit Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiration date ________ Signature ______________________

Name ______________________

Address ______________________

City ______________________ State ______ Zip ____________ 1/78

PLACE STAMP HERE

kilobaud

PETERBOROUGH NH 03458

Att: Subscriptions

books, etc.

- BASIC $4.95
- Basic Software Library, Vol I & II $24.95 ea., Vol III $39.95, Vol IV & V $9.95 ea. (formerly listed as "Scientific Research")
- Brand New Dictionary $15.95
- CMOS Cookbook $9.95
- Chemistry With a Computer $9.95
- Computer Dictionary $5.95
- Computer Programming Handbook $8.95
- Fortran Programming $7.95
- Fortran Workbook $3.95
- Fun With Computers and BASIC $6.95
- General Class Study Guide $5.95
- Hobby Computers Are Here $4.95
- Home Computers, 2^{10} Q&A, Vol I $7.95, Vol II $6.95
- An Introduction to Microcomputers Vol I $7.50, Vol II $12.50
- Introduction to Microprocessors $17.50 USA $20.00 elsewhere
- Microcomputer Dictionary $15.95
- Microcomputer Primer $7.95
- My Computer Likes Me . . . When I Speak BASIC $2.00
- The New Hobby Computers $4.95
- Novice Study Guide $4.95
- 1976 Periodical Guide for Computerists $3.00 January-June 1977 Edition $3.00
- A Quick Look at BASIC $4.95

kilobaud

- The New RTTY Handbook $5.95
- Scelbi's First Book of Computer Games $14.95
- Scelbi's Galaxy Game for the "6800" $14.95
- Secret Guide to Computers, Part I $2.75, Part II $2.50, Part III $3.50
- Some Common BASIC Programs $7.50
- Spencer's Computer Dictionary $5.95
- SSTV Handbook $5.00 (Soft), $7.00 (Hard)
- The Story of Computers $4.95
- TTL Cookbook $8.95
- TVT Cookbook $9.95
- Test Equipment Library $4.95 ea.
  - Vol I Component Testers
  - Vol II Audio Frequency Testers
  - Vol III Radio Frequency Testers
- Tychon's 8080 Hex Code Card $3.00
- Tychon's 8080 Octal Code Card $3.00
- The Underground Buying Guide $5.95
- VHF Antenna Handbook $4.95
- Weather Satellite Handbook $4.95
- What to Do After You Hit Return $8.00
- 60 Challenging Problems With BASIC Solutions $6.95
- 101 Games in BASIC $7.50
- 6800 Programming for Logic Design $7.50
- 6800 Software Gourmet Guide and Cookbook $9.95
- 8080 Programming for Logic Design $7.50
- 8080 Software Gourmet Guide and Cookbook $9.95
- KB Back Issues – Issue #1 – $5.00 ea.; Others – $3.00 ea.
- Binders for Kilobaud 1977 – $6.00 ea.; 2 for $11.00, $5.00 each additional binder.
- CM T-shirts $5.50 ea.


040 100 Pc
HEATHKIT

Jim-pak electronic components

One-Stop Component Center

QUALITY PRODUCTS

TTL

7400, 7402, 7404, 7408, 7410, 7420, 7430, 7432, 7442, 7447, 7473, 7474, 7475, 7476, 7485, 7490, 7492, 7493, 74107, 74121, 74122, 74123, 74125 (DM8093), 74154, 74161 (DM9316), 74176 (DM8280), 74177 (DM8281), 74192, 74193, 74367 (DM8097)

Low Power TTL Schottky

74LS00, 74LS02, 74LS04, 74LS08, 74LS10, 74LS20, 74LS30, 74LS32, 74LS73, 74LS74, 74LS75, 74LS83, 74LS85, 74LS86, 74LS90, 74LS109, 74LS123, 74LS136, 74LS138, 74LS151, 74LS175, 74LS367

C/MOS

4000, 4001, 4010, 4011, 4013, 4016, 4017, 4020, 4023, 4024, 4029, 4044, 4046, 4049, 4050, 4051, 4069, 4071, 4081, 4511

LINEAR

LM301V, LM307V, LM309K, LM311V, LM320K 5 (7905), LM320T-5 (7905), LM320T 12 (7912), LM320T-15 (7915), LM339N, LM340T-5 (7805), LM340T 12 (7812), LM340T-15 (7815), LM555V, LM556N, LM567V, LM723N, LM739N, LM741V, LM747N, LM1458V (5558V), LM1556V, LM3900N (CA3401)

MICROPROCESSOR

Z 80, 8080A, 8212, 8224, 8228, 6800, 6810, 6830, AY 5-1013, 2513/2140, 2101, 2102, 21L02, 7489, MM5262, 1702A, 82S23, 2708, DM8835N, N8T97

DIODES

IN751, IN4733, IN4734, IN4742, IN4744, IN4148 (IN914), IN4001, IN4004, IN4007, MDA-980-3

TRANSISTORS

C10681, 2N2222A, 2N2907A, MJE2955, MJE3055, 2N3055, 2N3904, 2N3906, 2N5129, 2N5139

RESISTORS

ASST 1, ASST 2, ASST 3, ASST 4, ASST 5, ASST 6, ASST 7

CAPACITORS

Aluminum Electrolytic

1mfd 50V, 4 7mfd 50V, 10mfd 50V, 22mfd 50V, 47mfd 50V, 100mfd 50V, 220mfd 50V, 470mfd 50V, 1000mfd 25V, 2200mfd 16V

Ceramic Disc

10pf 50V, 47pf 50V, 100pf 50V, 220pf 50V, 330pf 50V, 470pf 50V, 001mfd 50V, 0047mfd 50V, 01mfd 50V, 022mfd 50V, 047mfd 50V, 1mfd 50V

Dipped Tantalum

1mfd 35V, 22mfd 35V, 33mfd 35V, 47mfd 35V, 68mfd 35V, 1mfd 35V, 1 5mfd 35V, 2 2mfd 25V, 3 3mfd 25V, 4 7mfd 25V, 6 8mfd 25V, 10mfd 25V, 15mfd 25V, 33mfd 25V

SWITCHES

SPST 4 sta dip, SPST 8 sta dip, SPDT on-off-on toggle, SPDT on-none on toggle, DPDT on-off-on toggle, SPDT push button, SPDT push button, SPST push button, SPST push button, SPST push button, DPDT slide, DPDT on-none-on toggle

SOCKETS

8 pin low profile, 14 pin low profile, 16 pin low profile, 24 pin low profile, 40 pin low profile, Molex Pins, 14 pin wire wrap, 16 pin wire wrap, 24 pin wire wrap, 40 pin wire wrap, TO-3 Socket, TO-5 Socket

Display LEDS Discrete

5x7 dot matrix, 300 com anode, 300 com cathode, 600 com anode, 600 com cathode, 200LED Lamp-red

CONNECTORS

DB 25P Plug, DB25S Socket

DATA BOOKS*

7400/74LS Data Book
CMOS/Linear Data Book
Microprocessor/LED Data Book
*JIM-PAK Products only

A component product line developed for the independent dealer. Guaranteed, nationally advertised products. Complete **JIM-PAK** program includes national advertising, direct mail programs, store display racks, stock rotation plan and return policy. For dealer information, write or call **JIM-PAK**, 1021 Howard Avenue, San Carlos, CA 94070 (415) 592-8097.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

SEE YOUR LOCAL Jim-pak DEALER TODAY...

ALABAMA
- *Mobile* — *Lafayette Radio Electronics*

CALIFORNIA
- *Berkeley* — *Al Lasher Electronics*
- *Monterey* — *Zackit*
- *Palo Alto* — *Zack Electronics*
- *Pasadena* — *Dow Radio Inc.*
- *Sacramento* — *The Radio Place*
- *Sacramento* — *Zackit*
- *San Carlos* — *J&H Outlet Store*
- *San Francisco* — *Zack Electronics*
- *San Jose* — *Sunnyvale Electronics*
- *Vallejo* — *Zackit*
- *Walnut Creek* — *Byte Shop Computer Store*

CANADA
- *Alberta (Calgary)* — *The Computer Shop*

CONNECTICUT
- *Bridgeport* — *Computer World*

FLORIDA
- *Orlando* — *Altair Computer Center of Orlando*
- *Tampa* — *AMF Electronics*
- *Tampa* — *Microcomputer Systems*

FRANCE
- *Paris* — *Computer Boutique*

GEORGIA
- *Atlanta* — *Atlanta Computer Mart*

HAWAII
- *Aiea* — *Delcoms Hawaii*
- *Honolulu* — *Integrated Circuit Supply*

IDAHO
- *Idaho Falls* — *Audiotronics*

ILLINOIS
- *Evanston* — *Itty Bitty Machine Co.*
- *Groveland* — *Moyer Electronics*
- *Mount Prospect* — *Tri-State Electronics*
- *Oak Park* — *Spectronics Inc.*

INDIANA
- *East Chicago* — *Acro Electronics*
- *Hammond* — *Quantum Computer Works*

LOUISIANA
- *Baton Rouge* — *Davis Electronics Supply*

MARYLAND
- *Baltimore* — *Computer Workshop of Baltimore*
- *Baltimore* — *Everything Electronics*
- *Rockville* — *Computer Workshop*
- *Towson* — *Computers, Etc.*

MASSACHUSETTS
- *Medford* — *Tufts Electronics*
- *North Adams* — *Electronics Service Center*
- *Waltham* — *Computer Mart Inc.*

MICHIGAN
- *Lansing* — *Fulton Radio Supply*

MINNESOTA
- *Duluth* — *Northwest Radio of Duluth*
- *Eagan* — *Dacom Amateur Radio Center*

MISSOURI
- *Parkville* — *Computer Workshop of Kansas City*

MONTANA
- *Billings* — *Conley Radio Supply*

NEBRASKA
- *Lincoln* — *Altair Computer Center*
- *Omaha* — *Omaha Computer Store*

NEVADA
- *Las Vegas* — *Century 23*

NEW JERSEY
- *Hoboken* — *Hoboken Computer Works*
- *Ramsey* — *Typetronic Computer Store*

NEW YORK
- *Albany* — *Fort Orange Electronics*
- *New York* — *The Computer Stores Inc.*
- *New York* — *Computer Mart of New York*
- *Troy* — *Trojan Electronics*
- *White Plains* — *The Computer Corner*

NORTH CAROLINA
- *Raleigh* — *Byte Shop of Raleigh*

OHIO
- *Bucyrus* — *Mead Electronics*
- *Cincinnati* — *Digital Design*
- *Steubenville* — *Hosfelt Electronics*

OKLAHOMA
- *Guymon* — *Sound Service*
- *Oklahoma City* — *Bits, Bytes & Micros*

OREGON
- *Beaverton* — *Altair Computer Center*
- *Coos Bay* — *Herrick Electronix*
- *Ontario* — *Miller Electronics*

PENNSYLVANIA
- *Hershey* — *Microcomputer Systems Inc.*
- *Murraysville* — *Computer Workshop of Pittsburgh*

RHODE ISLAND
- *Cranston* — *Jabbour Electronics City*
- *Pawtucket* — *Jabbour Electronics City*

SOUTH CAROLINA
- *No. Charleston* — *Technical Services Inc.*

TENNESSEE
- *Memphis* — *Sere-Rose & Spencer Electronics*
- *Oak Ridge* — *Computer Denn*

TEXAS
- *Dallas* — *Computer Shops Inc.*
- *Houston* — *Altair Computer Center*
- *Houston* — *Interactive Computers*
- *San Antonio* — *Sherman Electronics Supply*

VIRGINIA
- *Alexandria* — *Computer Hardware Store*
- *Alexandria* — *Computers Plus*
- *Richmond* — *Computers-To-Go*
- *Springfield* — *Computer Workshop of North Virginia*

WASHINGTON
- *Bellevue* — *Altair Computer Center*
- *Longview* — *Progress Electronics*
- *Pasco* — *Riverview Electronics*
- *Spokane* — *Personal Computers*

WEST VIRGINIA
- *Morgantown* — *The Computer Corner*
- *Morgantown* — *Electro Distributing Co.*